

IMPLEMENTASI PERMAINAN PEMADAM KEBAKARAN MENGUNAKAN TEKNOLOGI VIRTUAL REALITY

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nur Muhammad Rashid

NIM: 115060807111038



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

IMPLEMENTASI PERMAINAN PEMADAM KEBAKARAN MENGGUNAKAN
TEKNOLOGI VIRTUAL REALITY

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Nur Muhammad Rashid

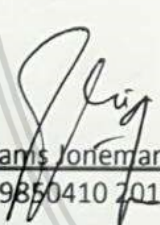
NIM: 115060807111038

Skripsi ini telah diuji dan dinyatakan lulus pada
31 Juli 2018

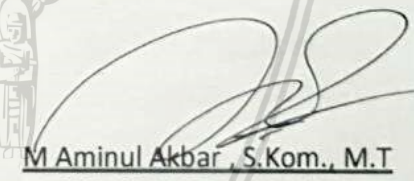
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Eriq M Adams Jorremaro, S.T, M.Kom

NIP: 19850410 201212 1 001

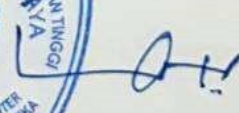

M Aminul Akbar, S.Kom., M.T

NIK: 2016078910131001

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 31 Juli 2018



Nur Muhammad Rashid

NIM: 115060807111038

KATA PENGANTAR

Puji syukur kehadiran Tuhan YME yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul **“IMPLEMENTASI PERMAINAN PEMADAM KEBAKARAN MENGGUNAKAN TEKNOLOGI VIRTUAL REALITY”** ini dapat terselesaikan. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Eriq Muhammad Adams Jonemaro, S.T, M.Kom dan Bapak Muhammad Aminul Akbar , S.Kom., M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Aswin Suharsono, S.T, M.T selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
3. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Seluruh dosen Fakultas Ilmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
5. Suharto, ST dan seluruh petugas damkar kota malang yang telah membantu menyediakan data untuk menyelesaikan skripsi ini.
6. Ayah dan Ibu dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
7. M. Daud Paweroy, S.T, yang telah membantu sejak SMA.
8. Billy Febiar, S.Kom, Herdian Adi W, S.Kom, Alfian Mukmin A, S.Kom, Triando Simarmata, S.Kom, yang telah memberikan pengalaman dan menemani selama masa kuliah.
9. Teman-teman “Miracle” (Andryanto, S.Kom, Sheila Zivana L, S.Kom, Alvin Hermawan, S.Kom, I Putu Yoga P., S.Kom, Arik Achmad E., S.Kom, Grandis Mahendra W.W., S.Kom, Weni Prameswari, S.Kom, Nadia Previani, S.Kom, Dwi Hardyanto, S.Kom, Afi Mufthihul, S.Kom, Dwi Vendy P., S.Kom, Anas Rachmadi., S.Kom, Rosikhan M., S.Kom, Claudio F.S., S.kom, Fendy Gusta Pradana, S.Kom, Albilaga L.P, S.Kom) yang selalu memberikan semangat, bantuan, dan dukungan penuh dalam masa perkuliahan dan pengerjaan skripsi.
10. Laqma Dica F, S.Kom yang pernah memberi semangat dan motivasi selama masah kuliah.
11. Randi Dwi N, S.Kom, yang telah membantu dan memberi ilham dalam menyelesaikan skripsi ini.

12. Teman-teman Raion Studio dan Raion Community yang sudah membagi pengalaman berorganisasi dan bekerja selama ini.
13. Ade Suluh dan Kingkeng, yang telah menemani pengerjaan skripsi ini tanpa mengenal lelah.
14. Teman-teman kontrakan lumba-lumba 19 memberi semangat dan banyak membantu penulis.
15. Seluruh Civitas Akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
16. Teman-teman seperjuangan Angkatan 2011 Informatika, terimakasih atas segala bantuannya selama menempuh studi di FILKOM Universitas Brawijaya.
17. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 31 Juli 2018

Penulis
ced19@gmail.com



ABSTRAK

Kebakaran merupakan bencana yang sering terjadi di lingkungan masyarakat namun masih banyak masyarakat awam yang belum mengetahui cara penggunaan serta jenis- jenis dari alat pemadam api ringan atau APAR. Menurut hasil wawancara dengan kepala UPT Pemadam Kebakaran, dalam pelatihan maupun pengenalan Pihak Dinas Pemadam Kebakaran dan Badan Penanggulangan Bencana Daerah (BPBD) terdapat kekurangan seperti jumlah APAR yang berjumlah sedikit sehingga karyawan pelatihan tidak dapat mencoba. Pelatihan APAR juga mempunyai resiko terjadinya kecelakaan. Oleh karena itu, studi kasus pada penelitian ini adalah masyarakat umum yang belum mengerti tentang jenis APAR beserta kegunaannya. Berdasarkan masalah yang telah dijelaskan, penelitian ini akan merancang dan membangun suatu simulasi dan permainan menggunakan teknologi *Virtual Reality*. Dengan memanfaatkan teknologi ini pengguna tidak diharuskan berhadapan dengan api secara nyata. Simulasi ini dapat meminimalisir kecelakaan saat pelatihan APAR serta menghemat penggunaan APAR. Simulasi ini dapat memvisualisasikan kejadian secara nyata sehingga pengguna tetap dapat merasakan bagaimana cara memadamkan api dan penggunaan alat pemadam api ringan atau APAR. Dengan adanya simulasi ini pengguna akan lebih siap saat kebakaran terjadi di lingkungannya. Kemudian untuk pengujian usabilitas dari implementasi dalam membangun Simulasi dan Permainan Pemadam Kebakaran ini didapatkan hasil yang memuaskan dengan rata-rata standard yaitu 80.

Kata kunci: Alat Pemadam Api Ringan, Permainan, Simulasi, Usabilitas, *Virtual Reality*.

ABSTRACT

Fire is a disaster that often occurs in the community but there are still many lay people who do not know how to use and the types of fire extinguishers or APAR. According to interviews with the head of the Technical Implementation Unit of Fire Brigade, in the training and the introduction of the Party Fire Department and Regional Disaster Management Agency (BPBD) there are shortcomings such as the number of APAR that amounts to a little, so employees can not try. APAR training also has an accident risk. Therefore, case studies in this study are the general public who do not understand about the type of APAR and its usefulness. Based on the problems described, this research will design and build a simulation and game using Virtual Reality technology. By utilizing this technology the user is not required to deal with real fire. This simulation can minimize accidents during APAR training and save the use of APAR. This simulation can visualize events so that the user can still feel how to extinguish the fire and the use of light extinguishers or APAR. With this simulation users will be better prepared when the fire occurred in the environment. Then for testing the usability of the implementation in building the Simulation and Fire Extinguisher game is obtained satisfactory results with an average above the standard that is 80.

Keywords: Game, Light Extinguishers, Simulation, Usability, Virtual Reality

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Alat Pemadam Api Ringan (APAR)	4
2.2 Jenis Kebakaran	5
2.3 <i>Virtual Reality</i>	6
2.3.1 Aplikasi <i>Virtual Reality</i>	6
2.3.2 Peranti <i>Virtual Reality (Virtual Device)</i>	6
2.4 Animasi 3D	7
2.5 Unity 3D	7
2.6 Bahasa Pemrograman C#	7
2.7 MDA Framework.....	7
2.8 Pengujian Perangkat Lunak.....	8
2.8.1 <i>White Box Testing</i>	8
2.8.2 <i>Black Box Testing</i>	8
2.9 <i>System Usability Scale (SUS)</i>	8
BAB 3 METODOLOGI	10

3.1 Studi Literatur	10
3.2 Analisis Kebutuhan	11
3.3 Perancangan	11
3.4 Implementasi	11
3.5 Pengujian	11
3.6 Analisis	12
BAB 4 Perancangan dan Implementasi	13
4.1 Perancangan Antarmuka Game	13
4.2 Perancangan Game	15
4.2.1 Deskripsi Game	15
4.2.2 <i>Player</i>	16
4.2.3 <i>Goal</i>	16
4.2.4 <i>Rules</i>	16
4.2.5 <i>Level</i>	16
4.2.6 <i>Resources dan Resources Management</i>	17
4.2.7 <i>Game Screen Flow</i>	17
4.2.8 <i>Information</i>	18
4.2.9 <i>Sequencing</i>	18
4.2.10 <i>Theme</i>	18
4.2.11 <i>Game as System</i>	19
4.3 Implementasi Game	19
4.3.1 Batasan Implementasi	19
4.3.2 Spesifikasi Perangkat Keras	19
4.3.3 Spesifikasi Perangkat Lunak	20
4.3.4 Implementasi Gameplay	20
4.4 Perancangan Pengujian	31
BAB 5 PENGUJIAN	33
5.1 Hasil Pengujian Menggunakan <i>White Box</i>	33
5.2 Hasil Pengujian Menggunakan <i>Black Box</i>	43
5.3 Pengujian Usabilitas	46
5.3.1 Perhitungan Skor Kuesioner	48
5.4 Analisis	48

5.4.1 Analisis Pengujian Dengan <i>White Box Testing</i>	48
5.4.2 Analisis Pengujian Dengan <i>Black Box Testing</i>	48
5.4.3 Hasil Pengujian Usabilitas	49
BAB 6 KESIMPULAN	51
6.1 Kesimpulan.....	51
6.2 Saran	51
DAFTAR PUSTAKA.....	52



DAFTAR TABEL

Tabel 4.1 Method untuk menentukan jumlah bintang yang didapat (skor)	21
Tabel 4.2 Method <i>OnParticleCollision</i>	21
Tabel 4.3 Kode program utama untuk mengecek apabila <i>lifetime</i> api habis	22
Tabel 4.4 Kode program utama pada <i>First Person Controller</i>	22
Tabel 4.5 Kode program utama untuk mengaktifkan semprotan APAR.....	23
Tabel 4.6 Kode program utama untuk mengganti jenis semprotan APAR	24
Tabel 4.7 Kode program saat memanggil method <i>AddStereoControllerToCameras</i>	27
Tabel 4.8 Implementasi level 1	30
Tabel 4.9 Implementasi level 2	31
Tabel 5.1 Kasus Uji Prosedur Perhitungan Skor (jumlah bintang)	35
Tabel 5.2 Tabel Kasus Uji Perhitungan Timbangan.....	37
Tabel 5.3 Kasus Uji Prosedur Interaksi Pemain sebagai Kontrol Karakter.....	40
Tabel 5.4 Kasus Uji Prosedur Mengaktifkan dan Mengganti Jenis Semprotan APAR.	43
Tabel 5.5 Tabel Kasus Uji Pengujian Prosedur	45
Tabel 5.6 Kuisisioner Pengujian Usabilitas Simulasi dan Permainan	47
Tabel 5.7 Sampel Perhitungan dari Pengisian Kuisisioner.....	49

DAFTAR GAMBAR

Gambar 2.1 Warna Alat Pemadam Api Ringan (APAR).....	5
Gambar 4.1 Perancangan tampilan menu Home	13
Gambar 4.2 Perancangan tampilan menu Level.....	14
Gambar 4.3 Perancangan tampilan Gameplay	14
Gambar 4.4 Game <i>Screen Flow Main Menu</i>	17
Gambar 4.5 Game <i>Screen Flow In Game</i>	18
Gambar 4.6 Penambahan komponen (<i>script</i>) <i>GvrViewer</i> pada sebuah <i>game object</i>	26
Gambar 4.7 Penambahan komponen (<i>script</i>) <i>GvrHead</i> pada sebuah <i>game object</i>	27
Gambar 4.8 Interface Menu Home	28
Gambar 4.9 Interface Pilih Level	29
Gambar 4.10 Interface GamePlay	29
Gambar 5.1 Pseudocode Prosedur Menang	33
Gambar 5.2 <i>Flow Graph</i> Prosedur Perhitungan Skor (jumlah bintang)	34
Gambar 5.3 Pseudocode Prosedur Menghancurkan Objek Api	36
Gambar 5.4 <i>Flow graph</i> Menghancurkan Objek Api.....	36
Gambar 5.5 Pseudocode Prosedur Interaksi Pemain sebagai Kontrol Karakter ..	38
Gambar 5.6 <i>Flow Graph</i> Prosedur Interaksi Pemain sebagai Kontrol Karakter....	39
Gambar 5.7 Pseudocode Prosedur mengaktifkan dan mengganti jenis semprotan APAR.	41
Gambar 5.8 <i>Flow Graph</i> Prosedur mengaktifkan dan mengganti jenis semprotan APAR.	42
Gambar 5.9 Pengujian Test Flow Diagram pada simulasi	44

BAB 1 PENDAHULUAN

1.1 Latar belakang

Dinas Pemadam Kebakaran merupakan unit gawat darurat dari pemerintah untuk penanggulangan tragedi kebakaran dan bencana alam. Para anggota juga dilengkapi dengan pakaian anti panas dan alat alat lain seperti helm dan sepatu khusus untuk tugas memadamkan api. Dalam tugas untuk memadamkan api terdapat alat alat yang mempunyai kegunaan masing masing sesuai dengan kelas kebakaran (Direktorat Jenderal Pembinaan Pelatihan dan Produktivitas, 2009).

Kebakaran merupakan bencana yang sering terjadi di lingkungan masyarakat namun masih banyak masyarakat awam yang mengetahui cara penggunaan serta jenis-jenis dari alat pemadam api ringan atau APAR. Sedangkan pemadam kebakaran sendiri tidak selalu tepat waktu saat bencana ini terjadi. APAR sendiri merupakan suatu alat pemadam yang dapat dibawa dan digunakan oleh satu orang, beratnya berkisar antara 1 hingga 15kg, dan digunakan untuk memadamkan kebakaran tingkat awal (Teguh Hambudi, 2015).

Dari hasil wawancara dengan bapak Suharto sebagai kepala UPT Dinas Pemadam Kebakaran kota Malang bahwa Pihak Dinas Pemadam Kebakaran dan Badan Penanggulangan Bencana Daerah (BPBD) saat ini menggunakan pengenalan APAR melalui sosialisasi atau pelatihan pada perusahaan, badan pemerintahan serta sekolah. Dalam pelatihan APAR terdapat kekurangan seperti jumlah APAR yang berjumlah sedikit sehingga warga yang mengikuti pelatihan tidak dapat mencoba. Pelatihan APAR juga mempunyai resiko terjadinya kecelakaan.

Simulasi pemadam kebakaran ini diterapkan pada teknologi *Virtual Reality*. *Virtual Reality* (VR) adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan oleh komputer (Michael Heim, 1993). Sehingga pengguna dapat berlatih memadamkan api tanpa harus berinteraksi dengan api sungguhan.

Teknologi ini akan menipiskan batas antara apa yang nyata dan apa yang dihasilkan komputer sehingga kita dapat melihat, merasa dan mencium. Hal ini membuat VR berguna sebagai alat untuk membantu persepsi dan interaksi pengguna dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. Sedangkan simulasi sendiri adalah kegiatan yg menyerupai sesuatu yang nyata beserta hal di sekitarnya (Michael Heim, 1993).

Pada penelitian sebelumnya, Alexander J (2016) mengimplementasikan virtual reality pada unit gawat darurat. Penelitian ini menggunakan platform virtual reality sebagai cara baru untuk menciptakan skenario medis berisiko tinggi yang ditargetkan untuk keadaan darurat di kantor. Menurut hasil penelitian, pengujian usability kepada dokter dan staff mendapatkan nilai rata-rata skor 75 dari 10 responden yang menunjukkan bahwa penggunaan virtual reality tersebut mampu memuaskan dokter dan staff yang bekerja pada unit gawat darurat tersebut.

Penulis mempunyai gagasan yang dapat memenuhi kebutuhan informasi yang disebutkan sebelumnya. Gagasan ini adalah merancang sebuah simulasi pemadam

kebakaran menggunakan teknologi *Virtual Reality*. Dengan memanfaatkan teknologi ini pengguna tidak diharuskan berhadapan dengan api secara nyata sehingga mengurangi presentase kecelakaan saat pelatihan. Penggunaan teknologi *Virtual Reality* dapat menghemat penggunaan APAR. Simulasi ini dapat memvisualisasikan kejadian secara nyata sehingga pengguna tetap dapat merasakan bagaimana cara memadamkan api dan penggunaan alat pemadam api ringan atau APAR. Dengan adanya simulasi ini pengguna akan lebih siap saat kebakaran terjadi di lingkungannya.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang, maka dirumuskan permasalahannya yaitu:

1. Bagaimana desain dan implementasi permainan pengenalan pemadam kebakaran?
2. Bagaimana hasil pengujian fungsional dari permainan pengenalan pemadam kebakaran?
3. Bagaimana hasil pengujian usabilitas dari permainan pengenalan pemadam kebakaran yang di implementasikan terhadap pengguna?

1.3 Tujuan

Adapun tujuan dari penelitian simulasi sebagai berikut:

1. Mendesain dan membangun rancangan untuk permainan pemadam kebakaran.
2. Menguji kebutuhan fungsional dari permainan pemadam kebakaran.
3. Menguji usabilitas dari permainan pemadam kebakaran.

1.4 Manfaat

Adapun manfaat dari penelitian simulasi sebagai berikut:

1. Membantu dinas Pemadam Kebakaran dalam mengenalkan APAR kepada masyarakat.
2. Melihat manfaat bagi masyarakat setelah mencoba simulasi pemadam kebakaran.
3. Menguji usabilitas dari simulasi dan permainan pemadam kebakaran.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan lebih terfokus dan tidak terjadi pelebaran topik, maka penelitian dibatasi dalam hal:

1. Simulasi yang ditampilkan hanya penggunaan alat pemadam api ringan dan pemadaman api.
2. Simulasi hanya terbatas pada penggunaan APAR.
3. Ruang lingkup pada penelitian ini meliputi *Virtual Reality* dan simulasi 3D.
4. Pengujian usabilitas hanya dilakukan dari sisi pengguna.
5. Api yang digunakan dalam simulasi hanya api kelas A, B, dan C

1.6 Sistematika pembahasan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

Bab I Pendahuluan

Bab pendahuluan berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

Bab II Dasar Teori

Bagian ini membahas tentang dasar teori secara luas mengenai *VR*(*Virtual reality*) dan teori pendukungnya untuk membangun aplikasi simulasi dengan *VR* pada media komputer.

Bab III Metedologi Penelitian

Pada bab ini dijelaskan langkah-langkah yang dilakukan dalam perancangan, implementasi dan pengujian dari simulasi pemadam kebakaran yang menggunakan teknologi *Virtual Reality*.

Bab IV Perancangan dan Implementasi

Pada bab ini berisi perancangan dari permainan yang dibangun dan membahas Implementasi dari Simulasi pemadam kebakaran yang menggunakan teknologi *Virtual Reality* sesuai dengan perancangan sistem yang telah dibuat.

Bab VI Pengujian & Analisis

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan

Bab VII Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan pada tugas akhir ini serta saran-saran untuk pengembangan lebih lanjut.

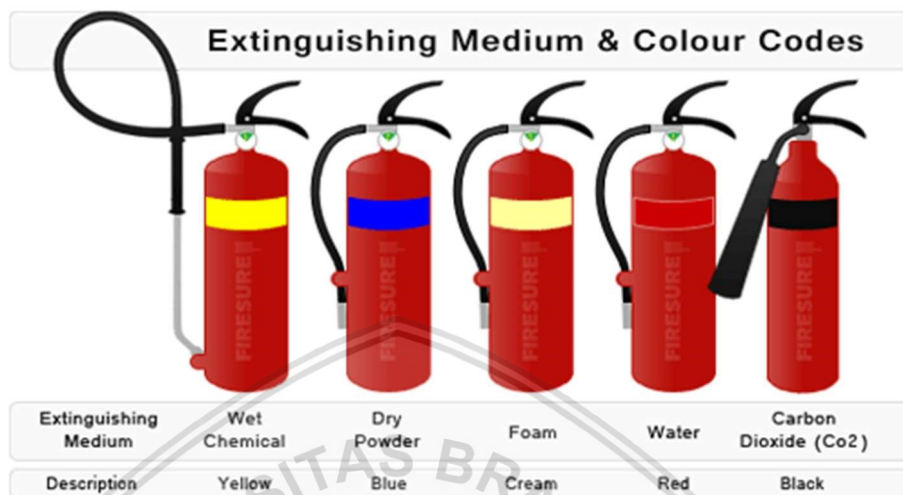
BAB 2 LANDASAN KEPUSTAKAAN

2.1 Alat Pemadam Api Ringan (APAR)

Alat pemadam api ringan adalah suatu alat pemadam yang dapat dibawa dan digunakan oleh satu orang, beratnya berkisar antara 1 hingga 15kg, dan digunakan untuk memadamkan kebakaran tingkat awal. Lama kerjanya berkisar antara 8 hingga 30 detik. APAR dibagi menjadi 5 jenis yang digolongkan untuk menangani kebakaran sesuai dengan kelas nya yang ditunjukkan pada Gambar 2.1.1.

- a. APAR jenis air (*Water Fire Extinguisher*)
Efektif untuk jenis api kelas A, seperti kebakaran pada kayu, kertas, kain, karet, dan plastic. Air merupakan salah satu bahan pemadam api yang paling berguna, sekaligus ekonomis. Semua pemadam api berbahan air produksi memiliki aplikasi tipe yang mampu menghasilkan arus yang terkonsentrasi, sehingga membuat operator mampu melawan api dari jarak yang lebih jauh daripada *nozzle* semprot biasa.
- b. APAR jenis tepung kimia (*Dry Chemical Powder*)
Efektif untuk jenis api kelas A (kayu, kertas, kain, karet, dan plastik), kelas B (bensin, gas, oli, cat, solvents, methanol, dan propane), serta kelas C (komputer, panel listrik, genset, dan gardu listrik). APAR yang berbahan bubuk kering sangat serba guna untuk mengatasi segala kelas api, serta cocok untuk mengatasi resiko tinggi. APAR jenis ini juga dapat digunakan untuk kebakaran pada kendaraan.
- c. APAR jenis Busa (*Foam Liquid AFFF*)
Efektif untuk jenis api kelas A dan B. Alat pemadam berbahan busa memiliki kemampuan untuk mengurangi resiko api kembali menyala setelah pemadaman. Setelah api dipadamkan, busa secara efektif menghilangkan uap, bersamaan dengan pendinginan api. APAR jenis busa sangat efektif terhadap bensin dan cairan yang mudah menguap, membentuk segel api diatas permukaan, dan mencegah pengapian ulang. Ideal untuk penggunaan multirisiko.
- d. APAR jenis CO₂ (*Carbon Dioxide*)
Alat pemadam api berbahan CO₂ sangat cocok untuk peralatan berlistrik dan api kelas B. Kemampuan tingginya yang tidak merusak serta efektif dan bersih yang sangat dikenal luas. CO₂ memiliki sifat non-konduktif dan antistatis. Oleh karena gas ini tidak berbahaya untuk peralatan dan bahan yang halus, sangat ideal untuk lingkungan kantor yang modern, dimana minyak, solvent dan lilin sering digunakan. Kinerja yang tidak merusak dan sangat efektif serta bersih sangatlah penting. Kedua model memiliki corong yang tidak berpenghantar dan *antistatis*, cocok untuk situasi yang melibatkan cairan yang mudah terbakar dan bahaya listrik
- e. APAR jenis *Hallon* (*Thermatic Halotron*)
Efektif untuk jenis api kelas A (kayu, kertas, kain, karet, dan plastik) dan kelas C (komputer, panel listrik, genset, dan gardu listrik). Alat pemadam

api ringan otomatis yang berisi clean agent halotron. Alat pemadam api ringan (APAR) otomatis ini menggunakan gas pendorong argon dan alat pengukur tekanan dipasang di alat pemadam api ringan (APAR) otomatis (Direktorat Jenderal Pembinaan Pelatihan dan Produktivitas, 2009).



Gambar 2.1 Warna Alat Pemadam Api Ringan (APAR)

2.2 Jenis Kebakaran

Menurut Peraturan Menteri Tenaga Kerja dan Transmigrasi Nomor 04/MEN/1980 Bab I Pasal 2 Ayat 1, kebakaran diklarifikasikan menjadi empat yaitu kategori A,B,C,dan D. Sedangkan *national fire protection association* (NFPA) menetapkan lima kategori jenis penyebab kebakaran, yaitu kelas A,B,C,D, dan K. Bahkan beberapa Negara menetapkan tambahan klasifikasi dengan kelas E (Direktorat Jenderal Pembinaan Pelatihan dan Produktivitas, 2009).

Klasifikasi tersebut sebagai berikut.

- Kebakaran Kelas A**
Kebakaran kelas A, yaitu kebakaran yang mengenai benda benda padat kecuali logam. Contohnya seperti kayu, kertas, kain, dan plastic. Alat atau media pemadam yang tepat untuk memadamkan kebakaran kelas ini adalah dengan pasir, tanah atau lumpur, tepung pemadam, busa, dan air
- Kebakaran Kelas B**
Kebakaran kelas B, yaitu kebakaran yang mengenai bahan bakar cair atau gas yang mudah terbakar. Contohnya seperti kerosene, solar, premium, LPG dan minyak goreng. Alat pemadam yang dapat dipergunakan pada kebakaran tersebut adalah tepung pemadam, busa, air dalam bentuk *spray*, atau kabut yang halus
- Kebakaran Kelas C**
Kebakaran kelas C, yaitu kebakaran instalasi listrik bertegangan. Contohnya seperti breaker listrik dan alat rumah tangga lainnya yang menggunakan listrik. Alat atau media pemadam yang tepat untuk

memadamkan kebakaran kelas ini adalah dengan karbondioksida (CO₂) dan tepung kering (*dry chemical*). Dalam pemadaman ini dilarang menggunakan air

d. Kebakaran Kelas D

Kebakaran kelas D, yaitu kebakaran yang mengenai benda logam padat. Contohnya seperti magnesium, aluminium, natrium, dan kalium. Alat atau media pemadam yang dipergunakan yaitu pasir halus dan kering serta dry powder khusus.

e. Kebakaran Kelas K

Kebakaran kelas K, yaitu kebakaran yang disebabkan oleh bahan akibat konsentrasi lemak yang tinggi. Kebakaran jenis ini banyak terjadi di dapur. Api yang timbul di dapur dapat dikategorikan sebagai api kelas B.

f. Kebakaran Kelas E

Kebakaran kelas E, yaitu kebakaran yang disebabkan oleh adanya hubungan arus pendek pada peralatan elektronik. Alat pemadam yang bisa digunakan untuk memadamkan kebakaran jenis ini adalah tepung kimia kering (*dry powder*). Namun alat ini memiliki kerusakan pada peralatan elektronik karena sifatnya yang lengket. Dengan demikian lebih cocok jika menggunakan pemadam api berbahan clean agent (Teguh Hambudi, 2015).

2.3 Virtual Reality

Virtual Reality merupakan gabungan dari dua kata yaitu virtual yang mempunyai arti efek tetapi tidak nyata, sedangkan reality adalah kejadian atau keadaan nyata. Sehingga saat kata digabungkan maka virtual reality merupakan kejadian atau keadaan yang menyerupai atau mempunyai efek yang nyata tetapi secara keseluruhan tidak nyata. Virtual Reality sendiri mengacu pada penggabungan dari objek dunia nyata ke dunia digital (Michael Heim, 1993).

2.3.1 Aplikasi Virtual Reality

Beberapa implementasi aplikasi Virtual Reality yaitu :

- Militer : *Flight simulation*, *battlefield simulation*, dan *virtual boot camp*.
- Pendidikan : *Virtual astronomy* dan *surgery simulation*.
- Kesehatan : *Human simulation* dan *virtual diagnostics*.
- Teknik : *Virtual rail construction* dan *virtual car design*.
- Bisnis : *Training simulation* dan bentuk 3d dari produk.
- Hiburan : Film dan *game*

2.3.2 Peranti Virtual Reality (Virtual Device)

Agar pengguna dapat merasakan dunia virtual secara nyata maka terdapat alat alat penunjang seperti ini.

- Glove* : hardware yang dapat mengetahui arah serta gerakan dari tangan pengguna. Setiap gerakan akan ditangkap oleh sensor lalu dikirim ke system VR.

- b. Headset : Alat yang digunakan agar pengguna dapat melihat dunia virtual dengan cara menengok ke samping atau belakang.
- c. Walker : alat yang digunakan agar pemain merasakan berjalan dalam dunia virtual. Alat ini akan menangkap setiap gerakan kaki penggunanya (Maria T Schultheis & Albert A Rizo, 2001).

2.4 Animasi 3D

Andy Beane mendefinisikan animasi 3D merupakan bagian dari virtual reality, 3D sendiri merupakan model palsu yang terlihat nyata. Dengan adanya animasi 3D maka pengguna yang melihat akan merasakan kejadian nyata. Animasi 3D sendiri merupakan bidang yang lebih besar cakupannya dari 3D *computer graphics*.

Animasi 3D digunakan di 3 industri utama yaitu *entertainment*, *Scientific*, dan lain lain. Masing masing industri menggunakan 3D secara berbeda dan hasilnya juga berbeda seperti film, video, visualisasi, *rapid prototyping* dan masih banyak yang lainnya. Tetapi animasi 3D masih berevolusi dan belum terlihat akhir dari cakupannya.

2.5 Unity 3D

Unity 3D merupakan salah satu dari game engine yang banyak digunakan. Unity tidak hanya dapat digunakan untuk membuat game di platform PC seperti playstation 4, wii, Xbox one, android, windowsphone dan lain sebagainya. Unity menggunakan bahasa pemrograman JAVA SCRIPT, CS SCRIPT (C#) dan BOO SCRIPT (Roedavan R, 2014).

Editor dan UI pada unity sangat mudah dipahami. Unity juga dapat mendukung semua format file. Unity secara rinci dapat digunakan untuk membuat video game 3D, real time animasi 3D dan visualisasi arsitektur dan isi serupa yang interaktif lainnya. Editor Unity dapat menggunakan plugin untuk web player dan menghasilkan game browser yang didukung oleh Windows dan Mac.

2.6 Bahasa Pemrograman C#

C# Merupakan sebuah Bahasa pemrograman yang sederhana, modern, berbasis objek, dan *type-safe*. C# berakar pada keluarga Bahasa C dan programmer C, C++ dan Java dapat familiar dengan cepat. C# distandarisasi oleh ECMA International sebagai standar ECMA-334 dan oleh ISO/IEC sebagai standar ISO/IEC 23270. Microsoft C# Compiler untuk .NET *Framework* merupakan implementasi yang sesuai dari kedua standar tersebut (Hejlsberg A, Torgersen M, Wiltamuth S, & Golde P, 2011).

2.7 MDA Framework

MDA Framework adalah konsep kerja untuk men-*design* sebuah *game*. Konsep MDA adalah *Mechanic*, *Dynamic* dan *Aesthetics* :

- a. *Mechanic* adalah sinonim untuk “*rule*”. Ini adalah bagaimana *game* berjalan dan apa saja batasan dari *game* tersebut. Apa yang pemain dapat lakukan kapan permainan berakhir dan apa yang terjadi ketika suatu tindakan diambil.
- b. *Dynamic* adalah gambaran bagaimana permainan dengan ketentuan *rule* yang sudah ada.
- c. *Aesthetics* adalah respon pemain ketika pemain memainkan permainan yang sudah diberi *rule* (Robin Hunicke, M. L. R. Z., 2015).

2.8 Pengujian Perangkat Lunak

Pengujian dapat berarti proses untuk mengecek apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai dengan standar tertentu. Secara umum ada dua jenis pengujian, yaitu Pengujian White Box dan pengujian Black Box

2.8.1 White Box Testing

Pengujian white box adalah pengujian yang bertujuan untuk menemukan kesalahan pada source code atau skema database. Pengujian white box memiliki beberapa metode pengujian. Salah satu metode pengujian white box adalah Control flow/Coverage Testing, pada metode ini test case dilakukan untuk memastikan bahwa semua statement, branch dan condition pada program telah dieksekusi paling tidak satu kali selama pengujian.

2.8.2 Black Box Testing

Pengujian black box atau behavioral testing berfokus pada persyaratan perangkat lunak. Dengan demikian, pengujian black-box memungkinkan perekrut perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian black-box bukan merupakan alternatif dari teknik white-box, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode white-box.

Pengujian black-box berusaha menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

2.9 System Usability Scale (SUS)

System Usability Scale (SUS) diciptakan oleh John Brooke pada tahun 1986. Temuan ini pada awalnya diciptakan sebagai skala yang “*quick and dirty*” untuk

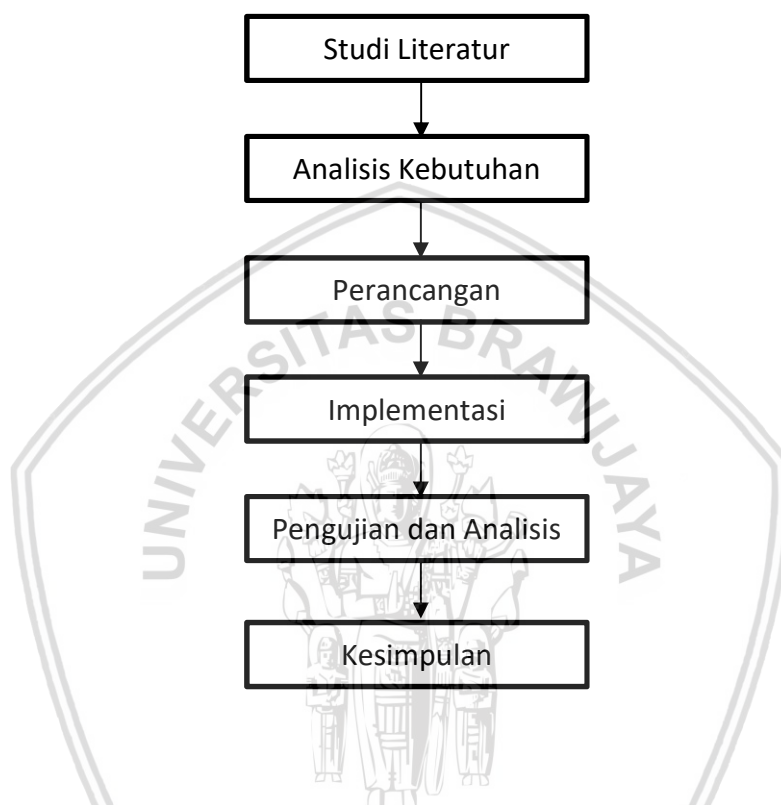
mengelola kegunaan tes pada sistem seperti aplikasi Terminal VT1000 (*"Green Screen"*) (Sauro, 2011). SUS adalah teknologi yang independen atau berdiri sendiri, oleh karena itu teknologi ini digunakan untuk menguji perangkat keras, perangkat lunak konsumen, situs web, telepon genggam dan masih banyak lagi. SUS terdiri dari 10 pernyataan dengan pilihan jawaban berupa angka 1 sampai 5 dari paling tidak setuju hingga paling setuju.

Kelebihan dari SUS adalah telah terbukti kehandalan dalam mendeteksi perbedaan pada ukuran sampel yang kecil kuesioner dan tersedia secara komersil atau bebas digunakan. SUS dapat digunakan pada sampel yang berukuran sangat kecil namun tetap menghasilkan hasil yang dapat diandalkan (Sauro, 2011). Kemudian SUS telah terbukti efektif membedakan antara sistem yang tidak dapat digunakan dan yang dapat digunakan serta dapat berkorelasi dengan kuesioner perhitungan penggunaan yang lain.



BAB 3 METODOLOGI

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari Desain dan Implementasi Simulasi Pemadam Kebakaran Menggunakan Teknologi Virtual Reality. Metodologi penelitian yang dilakukan melibatkan beberapa langkah :



3.1 Studi Literatur

Studi literatur merupakan penelusuran literatur yang bertujuan menyusun dasar teori yang digunakan untuk menunjang penyusunan tugas akhir ini. Teori-teori yang berkaitan dengan tugas akhir ini meliputi:

1. Pemadam Kebakaran
2. Virtual Reality
3. Alat Pemadam Api Ringan (APAR)
4. Animasi 3D
5. Unity 3D
6. Bahasa Pemrograman C#
7. MDA Framework
8. Pengujian Perangkat Lunak
 - a. White Box Testing

- b. Black Box Testing
- c. System Usability Scale

3.2 Analisis Kebutuhan

Kebutuhan fungsional yang nantinya akan disediakan oleh aplikasi ini antara lain adalah:

1. Aplikasi harus mampu menghitung skor yang didapatkan oleh pengguna.
2. Aplikasi harus mampu memadamkan api serta menghitung *lifetime* api.
3. Aplikasi harus mampu menjalankan gerak petugas pemadam kebakaran sesuai dengan kontrol dari user.
4. Aplikasi harus mampu mengaktifkan dan mengganti jenis semprotan APAR.

Sedangkan Kebutuhan Non Fungsional antara lain adalah:

1. Kebutuhan Hardware yang diperlukan untuk mengimplementasikan system.
2. Kebutuhan Software yang diperlukan untuk membuat system

3.3 Perancangan

Perancangan perangkat lunak dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Permainan Pemadam Kebakaran Menggunakan Teknologi Virtual Reality ini menggunakan MDA Framework.

3.4 Implementasi

Implementasi perangkat lunak mengacu kepada perancangan perangkat lunak. implementasi perangkat lunak dilakukan menggunakan bahasa pemrograman C#. Implementasi aplikasi ini antara lain:

- Penjabaran spesifikasi lingkungan pengembangan perangkat lunak.
- Implementasi basis data.
- Implementasi algoritma menggunakan pemrograman C# pada Unity 3D.
- Pembuatan user interface di dalam Unity.

3.5 Pengujian

Pada tahap ini penulis melakukan pengujian berdasarkan hasil implementasi yang telah dibangun. Tahapan ini bertujuan untuk mengetahui apakah aplikasi yang dibangun telah berjalan sesuai dengan spesifikasi dari kebutuhan yang melandasinya.

Sedangkan metode yang digunakan oleh penulis adalah metode pengujian *Black-box* dan *White-box*. Metode *black-box* berfokus pada persyaratan fungsional perangkat lunak seperti apakah interface aplikasi sudah berfungsi dengan baik, apakah aplikasi sudah dapat mengenali kontrol dari user . Sedangkan metode *white-box* lebih berfokus pada algoritma yang digunakan dalam perangkat lunak seperti menguji unit. Untuk pengujian usabilitas akan digunakan kuisioner dengan

metode SUS (*System Usability Scale*). Kuisisioner akan diberikan kepada responden terpilih yang telah menggunakan simulasi dan permainan pemadam kebakaran.

3.6 Analisis

Analisis atau evaluasi dilakukan setelah proses pengujian sistem selesai. Yang akan dianalisa adalah :

1. Menganalisis pembuatan simulasi pemadam kebakaran menggunakan teknologi virtual reality.
2. Menganalisis implementasi virtual reality pada simulasi pemadam kebakaran.
3. Menganalisis pengenalan simulasi pemadam kebakaran.



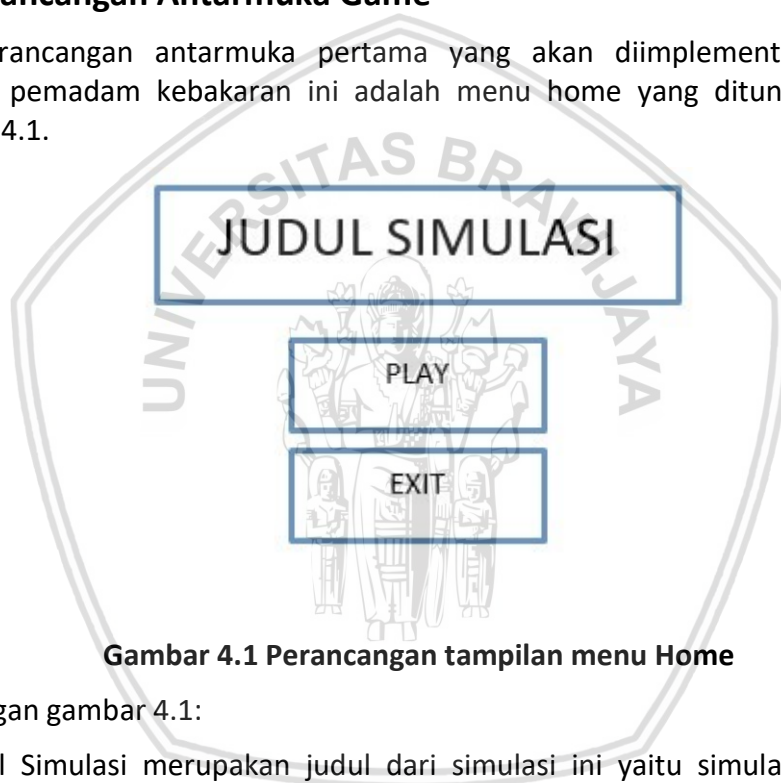
BAB 4

PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dibahas bagaimana perancangan dari permainan pemadam kebakaran menggunakan MDA Framework. Pada bab ini juga di perlihatkan tahapan dalam membuat permainan dari implementasi program hingga implementasi *level*.

4.1 Perancangan Antarmuka Game

Perancangan antarmuka pertama yang akan diimplementasikan pada simulasi pemadam kebakaran ini adalah menu home yang ditunjukkan pada Gambar 4.1.

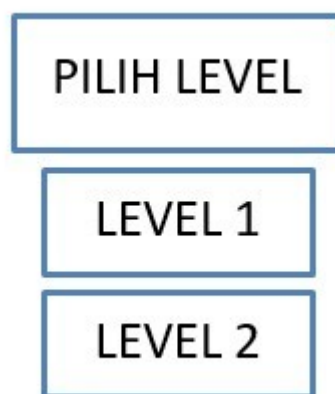


Gambar 4.1 Perancangan tampilan menu Home

Keterangan gambar 4.1:

1. Judul Simulasi merupakan judul dari simulasi ini yaitu simulasi pemadam kebakaran.
2. Play ditujukan untuk memulai simulasi.
3. Exit untuk keluar dari menu atau simulasi.

Setelah memilih tombol Play, maka akan ditampilkan menu untuk memilih level seperti yang ditunjukkan pada gambar 4.2.

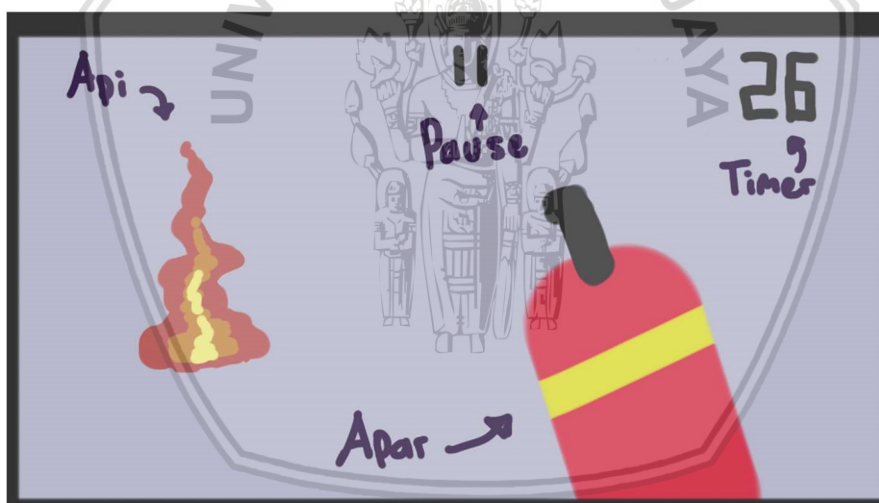


Gambar 4.2 Perancangan tampilan menu Level

Keterangan gambar 4.2:

1. Disini pemain akan dihadapkan pada pilihan level yang dibedakan sesuai tingkat kesulitan nya.

Kemudian tampilan gameplay akan ditunjukkan seperti pada gambar 4.3.



Gambar 4.3 Perancangan tampilan Gameplay

Keterangan gambar 4.3:

1. APAR merupakan objek simulasi yang digunakan untuk menyelesaikan simulasi.
2. Api adalah objek yang harus di padamkan menggunakan APAR.
3. Timer merupakan goal yang digunakan untuk membatasi waktu simulasi.
4. Pause untuk memberhentikan simulasi sejenak.

4.2 Perancangan MDA Framework

Dalam perancangan permainan pemadam kebakaran dilakukan menggunakan *MDA Framework*. MDA merupakan kependekan dari *Mechanics*, *Dynamics*, dan *Aesthetics*. *Mechanics* berisi *rules* dan algoritma yang dibuat oleh desainer game. Kumpulan dari *mechanics* yang bekerja akan mengakibatkan adanya *dynamics* atau *gameplay*. *Dynamics* secara otomatis memunculkan *aesthetics* atau respon dari *player* (Robin Hunicke, M. L. R. Z., 2015).

Tabel 4.1 MDA Framework pada Permainan

<i>Mechanics</i>	<i>Dynamics</i>	<i>Aesthetics</i>
Player, Api, APAR, Goal, Rules, Level, dan Information	Pemain dapat memilih jenis APAR untuk memadamkan api.	Pemain merasa tertantang saat memilih jenis APAR dan menyemprotkannya untuk mengurangi life time api sampai api padam.
	Pemain dapat secara bebas memilih api yang akan dipadamkan untuk menghemat waktu permainan.	Pemain merasa tertantang saat memadamkan api dengan waktu yang terbatas.

4.3 Perancangan Game

Dalam pembuatan sebuah game harus terdapat beberapa elemen agar game yang dimainkan terlihat menarik. Semua elemen tersebut akan dibahas pada sub bab dibawah ini.

4.3.1 Deskripsi Game

Game simulasi pemadam kebakaran ini merupakan game fps (first person shooter) yang diperuntukan bagi masyarakat agar mengetahui perbedaan jenis jenis pada apar.

Pemain berinteraksi menggunakan smartphone android yang dilengkapi google cardbox untuk melihat kedalam simulasi dan sebuah joystick untuk menggerakkan karakter. Game ini terbagi menjadi 2 level yang berbeda. Tiap level

dibedakan kesulitannya seperti jumlah api yang harus dipadamkan dan waktu simulasi berjalan.

4.3.2 Player

Dalam simulasi ini, pemain seolah olah dibuat dalam ruangan yang berisikan banyak api yang berbeda beda. Untuk memadamkan api, pemain harus mengarahkan cursor atau apar ke api lalu menyembrotkannya.

4.3.3 Goal

Pemain harus menyelesaikan simulasi dengan cara memadamkan api sesuai dengan jenis apar yang bisa digunakan untuk memadamkan api. Pemain harus bisa menyelesaikan simulasi sebelum waktu yang ditentukan habis.

Pada simulasi ini akan di terapkan rumus untuk mencapai goal sesuai kemampuan pemain untuk menyelesaikan simulasi dalam kurun waktu tertentu. Berikut rumus yang digunakan.

1. Bintang 3 akan didapat jika waktu yang tersisa $\geq 50\%$.
2. Bintang 2 akan didapat jika waktu yang tersisa $< 50\% - \geq 25\%$.
3. Bintang 1 akan didapat jika waktu yang tersisa $< 25\% - > 0\%$.
4. Pemain tidak akan mendapatkan bintang jika api tetap ada sampai waktu simulasi berakhir.

4.3.4 Rules

1. APAR air (Merah) untuk api jenis A
2. APAR tepung kimia (Biru) untuk api jenis A, B, dan C
3. APAR busa (Krim) untuk api jenis A dan B
4. APAR CO₂ (Hitam) untuk api jenis B
5. APAR Hallon (Hijau) untuk api jenis A dan C
6. Jika waktu melebihi batas maka pemain dianggap gagal
7. HP (Health Bar) api harus dikurangi sampai habis agar api hilang

4.3.5 Level

1. Level 1

Pada level 1 ini adalah level dimana pengguna harus memadamkan 9 api yang diantaranya terdiri dari 3 api A, 3 api B, dan 3 api C. Dalam level ini waktu yang ditentukan ialah 2 menit.

2. Level 2

Pada level 2 ini adalah level dimana pengguna harus memadamkan 15 api yang diantaranya terdiri dari 5 api A, 5 api B, dan 5 api C. Semua api ditempatkan

ditempat yang berbeda. Dalam level ini waktu yang ditentukan ialah 2 menit 30 detik.

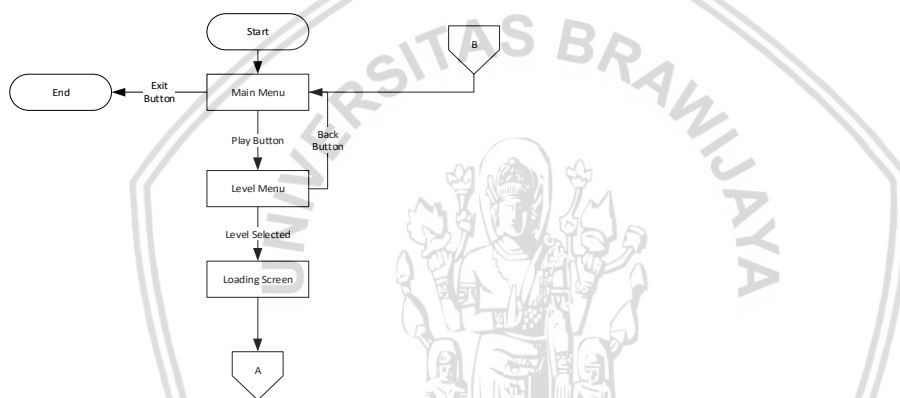
4.3.6 Resources dan Resources Management

Pemain tidak dapat keluar dari dinding yang merupakan *obstacle* atau batas agar pemain tidak keluar dari *stage* tersebut.

4.3.7 Game Screen Flow

Game screen flow merupakan urutan dari aktivitas dalam simulasi ini. Game screen digambarkan dengan persegi panjang sebagai suatu state dan arah anak panah untuk menggambarkan keadaan di dalam simulasi. Berikut game screen flow Simulasi Pemadam Kebakaran dapat dilihat pada Gambar 4.4.

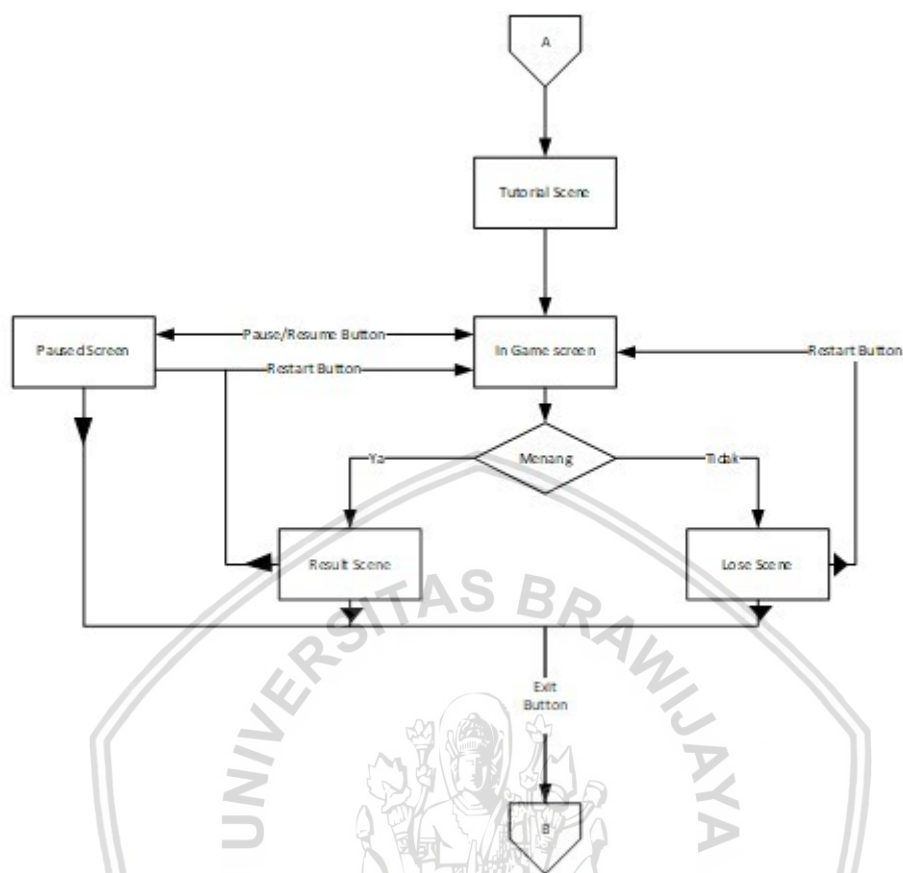
4.3.7.1 Game Screen Flow Main Menu



Gambar 4.4 Game Screen Flow Main Menu

Disaat pertama kali simulasi dijalankan, pemain akan masuk ke main menu dari simulasi. Dari menu terdapat 2 pilihan yang dapat diakses oleh pemain. Pertama adalah play button yang akan dilanjutkan ke level menu, dimana pemain bisa memilih 2 level yang berbeda. Kedua adalah exit button untuk membuat pemain keluar dari simulasi.

4.3.7.2 Game Screen Flow In Game



Gambar 4.5 Game Screen Flow In Game

Didalam gameplay simulasi pemadam kebakaran pemain harus memenangkan simulasi untuk dapat melihat result scene. Sedangkan jika pemain kalah maka akan muncul Lose Scene. Dalam gameplay, pemain dapat menjeda game (pause). Didalam ketiga scene tersebut terdapat restart button untuk memulai permainan dari awal. Sedangkan exit scene untuk kembali ke main menu.

4.3.8 Information

Informasi dalam simulasi akan diberikan sebelum permainan dimulai atau setelah memilih *level*. Informasi yang termasuk adalah info jenis api yang dapat dipadamkan sesuai jenis apar.

4.3.9 Sequencing

Simulasi ini akan berjalan cepat secepat yang diinginkan oleh pemain atau biasa disebut *real time*. Simulasi ini hanya bisa dimainkan oleh 1 orang saja.

4.3.10 Theme

Tema dalam simulasi ini merupakan warga atau seseorang yang harus memadamkan api yang terjadi secara tiba tiba di dalam gedung.

Pemain berinteraksi dengan joystick untuk menggerakkan karakter dan memanfaatkan teknologi *gyroscope* untuk menjalankan VR agar pemain dapat melihat dunia simulasi.

4.3.11 Game as System

Di dalam simulasi ini tidak terdapat subsistem di dalamnya.

4.4 Implementasi Game

Dalam tahap implementasi ini akan dilakukan penerapan dari perancangan sebelumnya. Pada subbab ini akan dijelaskan batasan implementasi dan implementasi *gameplay*.

4.4.1 Batasan Implementasi

Dalam subbab ini akan dijelaskan perangkat lunak serta perangkat keras yang digunakan dalam penerapan Simulasi Pemadam Kebakaran. Ada beberapa batasan dalam proses implementasi, berikut ini beberapa batasan tersebut.

1. Permainan hanya dikembangkan untuk smarthphone yang mempunyai fitur gyroscope.
2. Simulasi hanya bisa dimainkan menggunakan joystick Playstation 2 menggunakan kabel converter dan OTG (on the go).

4.4.2 Spesifikasi Perangkat Keras

Berikut ini merupakan lingkungan perangkat keras yang digunakan untuk mengembangkan simulasi Pemadam Kebakaran.

Perangkat keras untuk laptop.

1. Prosesor: Inter(R) Core(TM) i7-4710MQ CPU @2.50-GHz (8CPUs)
2. RAM: 8192 MB
3. Graphic Card: Nvidia Geforce GTX 850M
4. Kapasitas Hardisk: 1000 GB
5. Monitor: 14" HD

Perangkat keras untuk smartphone.

1. Prosesor: Octa-core (4x1.5 GHz Cortex-A53)
2. RAM: 3663 MB
3. Kapasitas Hardisk: 25448 MB
4. Monitor: 5,66"

Perangkat keras untuk joystick.

1. Sony Playstation 2 joystick
2. Kabel converter
3. Kabel OTG (on the go)

4.4.3 Spesifikasi Perangkat Lunak

Berikut ini merupakan lingkungan perangkat lunak yang digunakan untuk mengembangkan simulasi Pemadam Kebakaran.

Perangkat Lunak untuk laptop.

1. Sistem Operasi Windows 10 64 bit.
2. Unity 3D
3. Monodevelop

Perangkat lunak untuk smartphone

1. Android versi 6.0.1
2. Fitur gyroscope
3. Google VR sdk for unity versi 0.9.1

4.4.4 Implementasi Gameplay

Pada tahap ini akan dibahas tentang implementasi prosedur program dan implementasi dari user interface.

4.4.4.1 Implementasi Kode Program

Implementasi pada game ini melibatkan berbagai proses dan method program. Berikut ini merupakan proses dan method yang paling utama pada implementasi game ini.

1. Implementasi Perhitungan Skor

Pada implementasi perhitungan skor (penentuan jumlah bintang yang didapat) dilakukan perhitungan dengan menggunakan rumus yang telah ditentukan sebelumnya.

Tabel 4.1 Method untuk menentukan jumlah bintang yang didapat (skor)

```

public void setBintang(){
    int jumlahBintang = 0;
    float persenSisa = (timerremaining / DataStore.totalTime) * 100;
    if (persenSisa >= 50) {
        jumlahBintang = 3;
    } else if (persenSisa >= 25) {
        jumlahBintang = 2;
    } else if (persenSisa > 0) {
        jumlahBintang = 1;
    }

    if (jumlahBintang > 0) {
        panelgameOver.transform.Find("bintang
1").gameObject.SetActive(true);
    }
    if (jumlahBintang > 1) {
        panelgameOver.transform.Find("bintang
2").gameObject.SetActive(true);
    }
    if (jumlahBintang > 2) {
        panelgameOver.transform.Find("bintang
3").gameObject.SetActive(true);
    }
}

```

Method ini akan menampilkan 1 bintang apabila pemain masih memiliki sisa waktu ketika menyelesaikan game ini, kemudian akan menampilkan 2 bintang apabila sisa waktu tersebut lebih dari 25% dibandingkan total waktu yang diberikan dan akan menampilkan 3 bintang apabila sisa waktu lebih dari 50% dibandingkan total waktu yang didapat pada level yang dimainkan.

2. Implementasi Memadamkan Api

Pada implementasi memadamkan api terdapat dua bagian utama kode program yaitu pada saat semprotan APAR mengenai api (*collide*) dan ketika *lifetime* dari api telah habis (objek api dimatikan). Berikut adalah kode program utama pada saat semprotan APAR mengenai api (memanfaatkan method *OnParticleCollision*).

Tabel 4.2 Method *OnParticleCollision*

```

void OnParticleCollision(GameObject other)
{
    if (other.tag == "Spray")
    {
        if (!DataStore.isLifeTimeBarActive)
        {
            DataStore.isLifeTimeBarActive = true;
            DataStore.mainController.setUILifeBarActive(true);
        }

        if (TIPE_APAR.isInRule(this.tipeApi, DataStore.selectedAPAR))
        {
            this.lifeTimeValue -= 5f;
        }
    }
}

```

```

        else
        {
            this.lifeTimeValue -= 0.1f;
        }

        DataStore.lifeTimeBarValue = this.lifeTimeValue;
    }
}

```

Method `OnParticleCollision` ini dijalankan ketika particle dari semprotan APAR mengenai (*collide*) objek api, kemudian pada saat itu juga *lifetime* dari api akan dikurangi sesuai dengan efektifitas jenis APAR yang digunakan pada tipe sumber api yang disemprot. Kemudian berikut adalah kode program utama ketika *lifetime* dari objek api habis.

Tabel 4.3 Kode program utama untuk mengecek apabila *lifetime* api habis

```

void Update () {
    if (lifeTimeValue <= 0f)
    {
        DataStore.isLifeTimeBarActive = false;
        DataStore.mainController.setUILifeBarActive(false);
        DataStore.mainController.tambahJumlahApiPadam();
        this.gameObject.SetActive(false);
    }
}

```

Pada method `update` diatas dilakukan pengecekan sisa *lifetime* dari objek api. Apabila *lifetime* tersebut habis maka data jumlah api yang padam akan ditambahkan kemudian game objek api tersebut dimatikan.

3. Implementasi Interaksi Pemain sebagai Kontrol Karakter

Pada implementasi interaksi pemain sebagai kontrol karakter diterapkan kode program yang dapat mengecek masukan dari pemain melalui perangkat *joystick* untuk kemudian diubah menjadi respon dari karakter dalam *game*.

Kontrol untuk menjalankan karakter ke berbagai arah diimplementasikan dengan memanfaatkan *standard assets (script)* dari unity untuk game dengan tipe karakter *first person*. Berikut potongan kode program utama pada *script* tersebut.

Tabel 4.4 Kode program utama pada *First Person Controller*

```

private void FixedUpdate()
{
    float speed;

    GetInput(out speed);
    Vector3 desiredMove = transform.forward*m_Input.y +
    transform.right*m_Input.x;

    desiredMove = Vector3.ProjectOnPlane(desiredMove,
    hitInfo.normal).normalized;

    m_MoveDir.x = desiredMove.x*speed;
    m_MoveDir.z = desiredMove.z*speed;

    if (m_CharacterController.isGrounded)
    {

```

```

        m_MoveDir.y = -m_StickToGroundForce;

        if (m_Jump)
        {
            m_MoveDir.y = m_JumpSpeed;
            PlayJumpSound();
            m_Jump = false;
            m_Jumping = true;
        }
    }
    else
    {
        m_MoveDir += Physics.gravity*m_GravityMultiplier*Time.fixedDeltaTime;
    }
    m_CollisionFlags =
    m_CharacterController.Move(m_MoveDir*Time.fixedDeltaTime);

    ProgressStepCycle(speed);
    UpdateCameraPosition(speed);
}

private void GetInput(out float speed)
{
    float horizontal = Input.GetAxis("Horizontal");
    float vertical = Input.GetAxis("Vertical");

    bool waswalking = m_IsWalking;

    speed = m_IsWalking ? m_WalkSpeed : m_RunSpeed;
    m_Input = new Vector2(horizontal, vertical);

    if (m_Input.sqrMagnitude > 1)
    {
        m_Input.Normalize();
    }
}

```

Kemudian implementasi kode program untuk mengaktifkan semprotan APAR adalah sebagai berikut.

Tabel 4.5 Kode program utama untuk mengaktifkan semprotan APAR

```

void Update () {
    if (DataStore.isStarted) {
        updateTransform();
        if (!DataStore.isPaused && !DataStore.isGameOver && currentAPARContent > 0f)
        {
            if (GvrViewer.Instance.VRModeEnabled)
            {
                if ((Input.GetButton("Fire3") || Input.GetKey(KeyCode.Q) ||
                Input.GetButton(JOYSTICK_CODE.CROSS)))
                {
                    spray.enableEmission = true;
                    currentAPARContent -= DataStore.mainController.APARReduceSize;
                }
                else
                {
                    spray.enableEmission = false;
                }
            }
            else
            {
                if (Input.GetButton("Fire3"))
                {
                    spray.enableEmission = true;
                    currentAPARContent -= DataStore.mainController.APARReduceSize;
                }
            }
        }
    }
}

```

```

    }
    else
    {
        spray.enableEmission = false;
    }
}
if (currentAPARContent < 0)
{
    currentAPARContent = 0f;
    DataStore.APARContent[DataStore.selectedAPAR] = 0f;
}
currentAPARContentText.text = (int)currentAPARContent + "%";
}
else
{
    spray.enableEmission = false;
}
}
}

```

Selanjutnya implementasi kode program untuk mengganti jenis APAR yang sedang digunakan serta melakukan *pause* pada game adalah sebagai berikut.

Tabel 4.6 Kode program utama untuk mengganti jenis semprotan APAR

```

void Update () {
    if (!DataStore.isGameOver) {
        if (DataStore.isStarted)
        {
            if (timerremaining > 0)
            {
                if (!DataStore.isPaused)
                {
                    timerremaining -= Time.deltaTime;
                    timeText.text = getTimeFormat(timerremaining);
                }

                if (Input.GetKeyDown(KeyCode.Z) ||
Input.GetButtonDown(JOYSTICK_CODE.CUBE))
                {
                    if (!DataStore.isPaused)
                    {
                        changeAPAR();
                    }
                }
                else if (Input.GetKeyDown(KeyCode.P) ||
Input.GetButtonDown(JOYSTICK_CODE.L2))
                {
                    if (!DataStore.isPaused)
                    {
                        setPause();
                    }
                    else
                    {
                        resumeGame();
                    }
                }
            }
            updateApiLifeTimeBar();

            if (DataStore.isPaused)
            {
                checkInputPause();
            }
        }

        if (DataStore.checkAPAREmpty())
        {
            setGameOver(true);
        }
    }
}

```

```

    }

    if (timerremaining <= 0 || jumlahApiPadam >= DataStore.jumlahApi)
    {
        setGameOver(false);
    }
}
else if(isInTutorState)
{
    checkInputTutor();
}
}
else
{
    checkInputGameOver();
}
}

public void changeAPAR()
{
    /*Ganti/geser APAR switcer di UI

    APARSwitcher.Find(TIPE_APAR.getAPARName(DataStore.selectedAPAR)).Find("Active").gameObject.SetActive(false);

    APARSwitcher.Find(TIPE_APAR.getAPARName(DataStore.selectedAPAR)).Find("Content").gameObject.SetActive(false);
    int prevSelectedAPAR = DataStore.selectedAPAR;
    if (DataStore.selectedAPAR >= (TIPE_APAR.APAR_LIST.Length - 1))
    {
        DataStore.selectedAPAR = 0;
    }
    else
    {
        DataStore.selectedAPAR++;
    }
    sprayControl.changeAPAR(prevSelectedAPAR);
    APARSwitcher.Find(TIPE_APAR.getAPARName(DataStore.selectedAPAR)).Find("Content").GetComponent<Text>();

    APARSwitcher.Find(TIPE_APAR.getAPARName(DataStore.selectedAPAR)).Find("Active").gameObject.SetActive(true);

    APARSwitcher.Find(TIPE_APAR.getAPARName(DataStore.selectedAPAR)).Find("Content").gameObject.SetActive(true);

    /*Ganti material dan texture dari objek 3D APAR
    Material aparMat = Resources.Load("Materials/fire_extinguisher_" +
    TIPE_APAR.getAPARName(DataStore.selectedAPAR) + "", typeof(Material)) as Material;
    DataStore.extinguisher.GetComponent<MeshRenderer>().material = aparMat;
    }

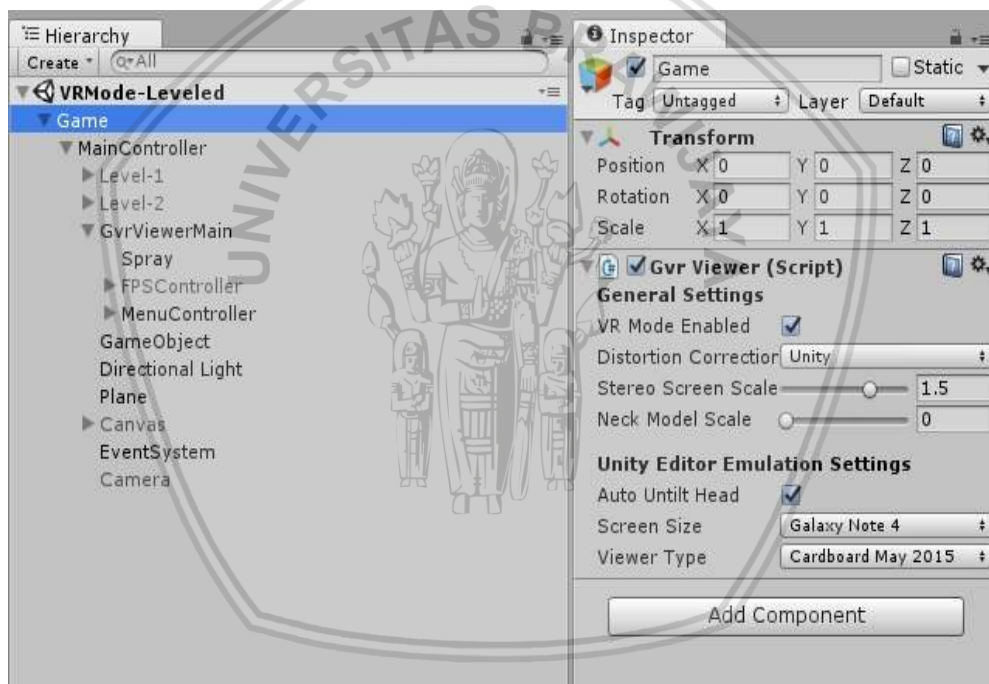
    public void setPause()
    {
        DataStore.isPaused = true;
        currentIndexButton = 1;
        pauseListButtons[0].transform.parent.gameObject.SetActive(true);
        EventSystem.current.SetSelectedGameObject(null);
        pauseListButtons[1].Select();
        FPSController.GetComponent<FirstPersonController>().enabled = false;
        extinguisher.SetActive(false);
    }
}

```

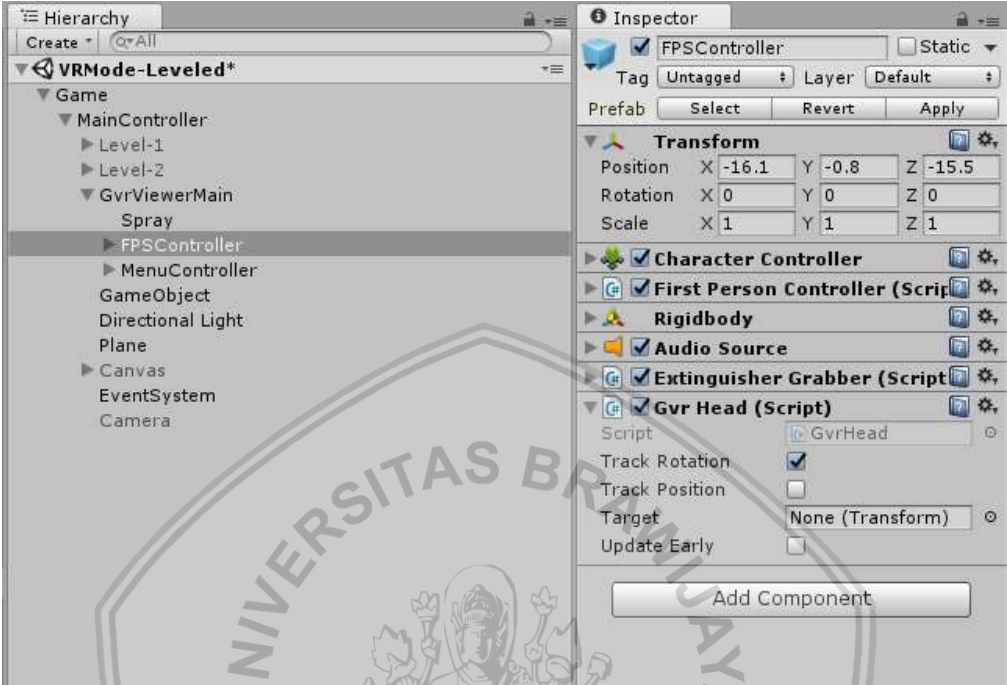

4. Implementasi Menampilkan Game dengan tampilan VR

Implementasi menampilkan game dengan tampilan *virtual reality* dilakukan dengan menggunakan *library* tambahan dari Google, yaitu Google VR SDK for Unity. Untuk mengubah tampilan game menjadi VR dengan *library* ini, cukup menambahkan komponen (*script*) bernama *GvrViewer* yang berasal dari *library* ini kedalam sebuah *game object* dan mengaktifkan *VR Mode* pada *script* ini.

Gambar 4.6 Penambahan komponen (*script*) *GvrViewer* pada sebuah *game object*



Kemudian dilakukan kembali penambahan komponen (*script*) yang berasal dari library ini bernama *GvrHead* pada game objek yang ingin dijadikan sebagai objek utama yang akan dirubah rotasinya pada saat VR digunakan sesuai dengan arah pandangan pemain.



Gambar 4.7 Penambahan komponen (*script*) *GvrHead* pada sebuah *game object*

Setelah kedua komponen (*script*) tersebut ditambahkan, maka selanjutnya dilakukan pemanggilan method *AddStereoControllerToCameras* dari library ini untuk menambahkan pengaturan *stereo* (dua tampilan) pada setiap objek *camera* yang sedang aktif pada saat game ini dijalankan. Berikut adalah potongan kode program untuk memanggil method tersebut yang dilakukan pada saat akan memasuki *gameplay* utama.

Tabel 4.7 Kode program saat memanggil method *AddStereoControllerToCameras*

```
public void goToLevel(int index)
{
    menuButton[0].transform.parent.gameObject.SetActive(true);
    levelButton[0].transform.parent.gameObject.SetActive(false);
    this.currentIndexButton = 0;
    isLevelState = false;
    this.gameObject.SetActive(false);
    DataStore.mainController.transform.Find("Level-" + index).gameObject.SetActive(true);
    DataStore.mainController.transform.Find("Level-" +
index).GetComponent<LevelController>().initLevel();
    DataStore.mainController.FPSController.SetActive(true);
    DataStore.mainController.OnLevelStarted();
    DataStore.levelSelected = index;
    GvrViewer.AddStereoControllerToCameras();
}
```

```
}
```

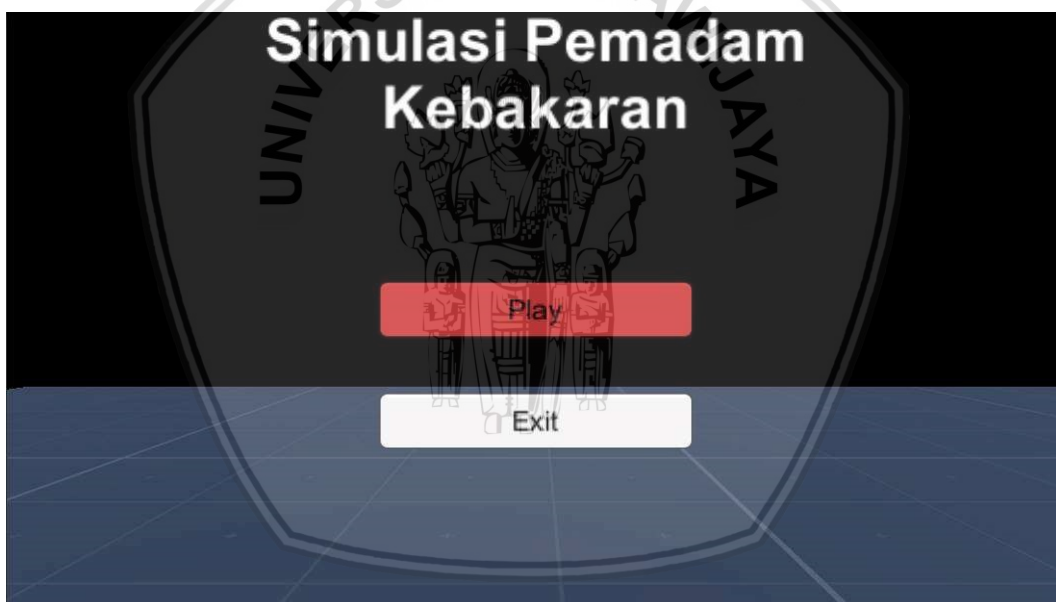
4.4.4.2 Implementasi User Interface

Pada tahap ini akan diperlihatkan desain interface dari Simulasi Pemadam Kebakaran. Pada simulasi ini terdapat beberapa tampilan seperti menu home, pilih level, dan gameplay.

Berikut tampilan interface dari menu home yang akan ditunjukkan pada gambar 4.8.

Keterangan gambar 4.8 adalah sebagai berikut:

1. Judul simulasi.
2. Icon untuk memulai simulasi.
3. Icon untuk keluar dari simulasi.



Gambar 4.8 Interface Menu Home

Setelah pengguna memilih icon play, maka akan ditampilkan interface untuk memilih level yang akan ditunjukkan pada gambar 4.9.



Gambar 4.9 Interface Pilih Level

Keterangan gambar 4.9 adalah sebagai berikut:

1. Pilihan level yang diinginkan pengguna dalam simulasi.

Setelah pengguna memilih level, maka akan ditampilkan interface gameplay yang akan ditunjukkan pada gambar 4.10.



Gambar 4.10 Interface GamePlay

Keterangan gambar 4.10 adalah sebagai berikut:

1. Lifetime atau nyawa dari api yang akan berkurang jika pengguna menyemprot api.
2. Jumlah api yang masih ada didalam simulasi.
3. Waktu yang tersisa untuk mencapai goal.
4. Jenis APAR dan sisa isi tabung.
5. Api yang harus dipadamkan.
6. APAR yang digunakan memadamkan api dengan cara menyemprotkannya pada api.

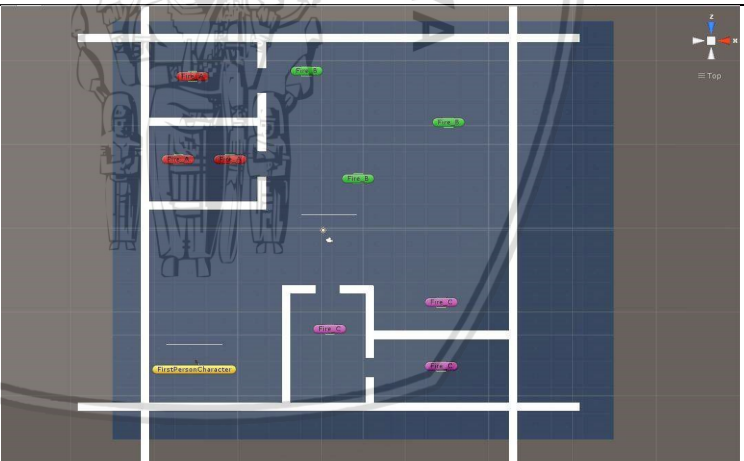
4.4.4.3 Implementasi Level

Pada subbab ini akan menjelaskan bagaimana implementasi *level design* untuk simulasi pemadam kebakaran. Akan ada 2 level yang tersedia pada simulasi ini.

1. Level 1

Level 1 akan dijelaskan pada Tabel 4.1 Implementasi level 1.

Tabel 4.8 Implementasi level 1

Level 1	
Keterangan	Pada level 1 ini adalah level dimana pengguna harus memadamkan 9 api yang diantaranya terdiri dari 3 api A, 3 api B, dan 3 api C. Dalam level ini waktu yang ditentukan ialah 2 menit.

2. Level 2

Level 2 akan dijelaskan pada Tabel 4.2 Implementasi level 2.

Tabel 4.9 Implementasi level 2

Level 2	
Keterangan	Pada level 2 ini adalah level dimana pengguna harus memadamkan 15 api yang diantaranya terdiri dari 5 api A, 5 api B, dan 5 api C. Semua api ditempatkan ditempat yang berbeda. Dalam level ini waktu yang ditentukan ialah 2 menit 30 detik.

4.5 Perancangan Pengujian

Dalam tahap ini akan dibahas pernacangan tahap demi tahap dalam pengujian yang akan dilakukan. Berikut ini adalah beberapa pengujian nya.

1. Pengujian White Box

Dalam Pengujian White Box akan digunakan pengujian dengan Teknik basis path testing.

2. Pengujian Black Box

Dalam Pengujian Black Box testing akan digunakan metode TFD atau Test Flow Diagram. TFD akan digunakan sebagai gambaran behavior dari sudut pandang player. Metode ini dibuat dengan melakukan pendekatan formal untuk testing sebuah desain.

3. Pengujian Usabilitas

Pengujian Usabilitas berguna untuk mengevaluasi tingkat dimana responden dapat berinteraksi secara efektif dengan simulasi dan sejauh mana simulasi ini dapat membimbing tindakan dari responden, memberikan umpan balik yang berarti dan menerapkan pendekatan pada interaksi yang konsisten. Berikut pernyataan nya.

1. Saya merasa dapat menggunakan simulasi ini dengan sangat percaya diri.

2. Saya perlu belajar banyak hal sebelum menggunakan simulasi ini.
 3. Menurut saya, kontrol dan keseluruhan dari simulasi ini mudah digunakan.
 4. Menurut saya, terdapat kerumitan yang tidak penting didalam simulasi ini.
 5. Menurut saya, simulasi ini dapat membantu saya menggunakan APAR pada lingkungan disekitar saya.
 6. Menurut saya, banyak fungsi pada simulasi ini berjalan secara tidak konsisten.
 7. Menurut saya, setiap fungsi pada simulasi ini dapat berjalan dengan sangat baik.
 8. Menurut saya, saya membutuhkan bantuan dari teknisi untuk menggunakan simulasi ini.
 9. Menurut saya, kebanyakan orang akan belajar dengan cepat dalam menggunakan aplikasi ini.
 10. Menurut saya, simulasi ini mempunyai penggunaan yang sangat rumit.(Peres, S & Pham, 2013)
4. Pengujian Dengan Black Box Testing

Dalam Pengujian Black Box testing akan digunakan metode TFD atau Test Flow Diagram. TFD akan digunakan sebagai gambaran behavior dari sudut pandang player. Metode ini dibuat dengan melakukan pendekatan formal untuk testing sebuah desain.

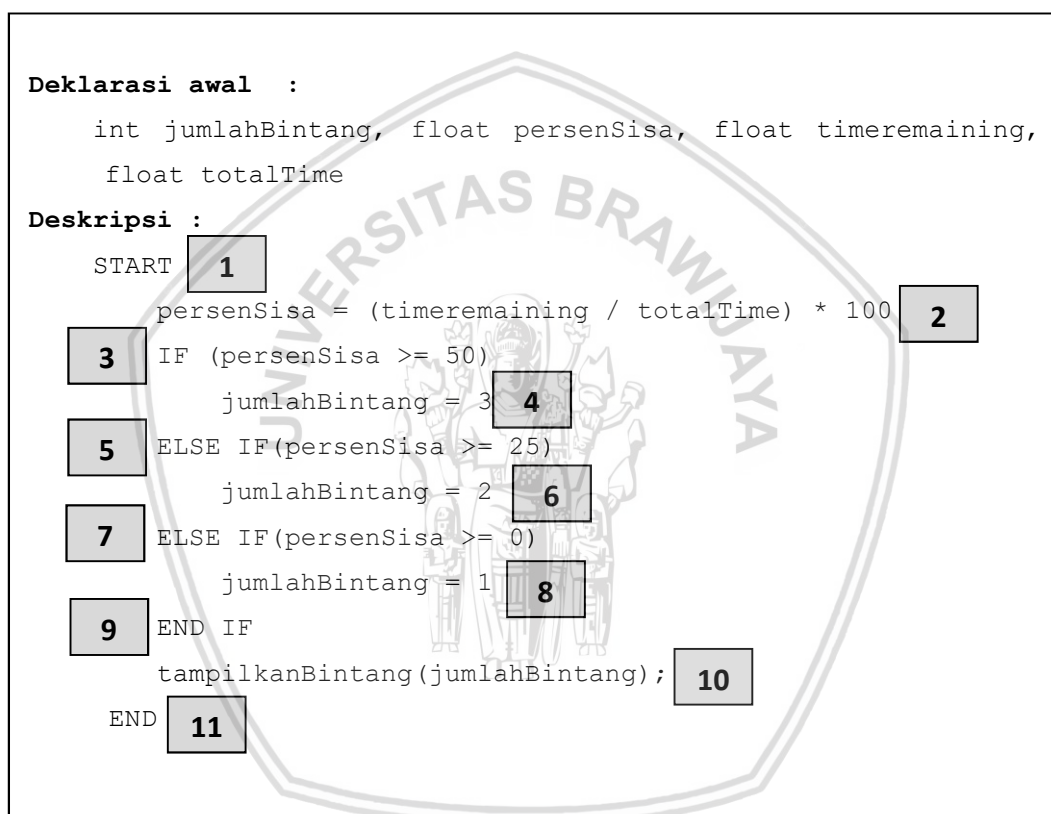
BAB 5 PENGUJIAN

Pada bab ini akan dibahas tentang hasil dari rancangan pengujian simulasi yang dilakukan terhadap 20 masyarakat. Metode yang digunakan adalah black box testing, dan pengujian usabilitas.

5.1 Hasil Pengujian Menggunakan *White Box*

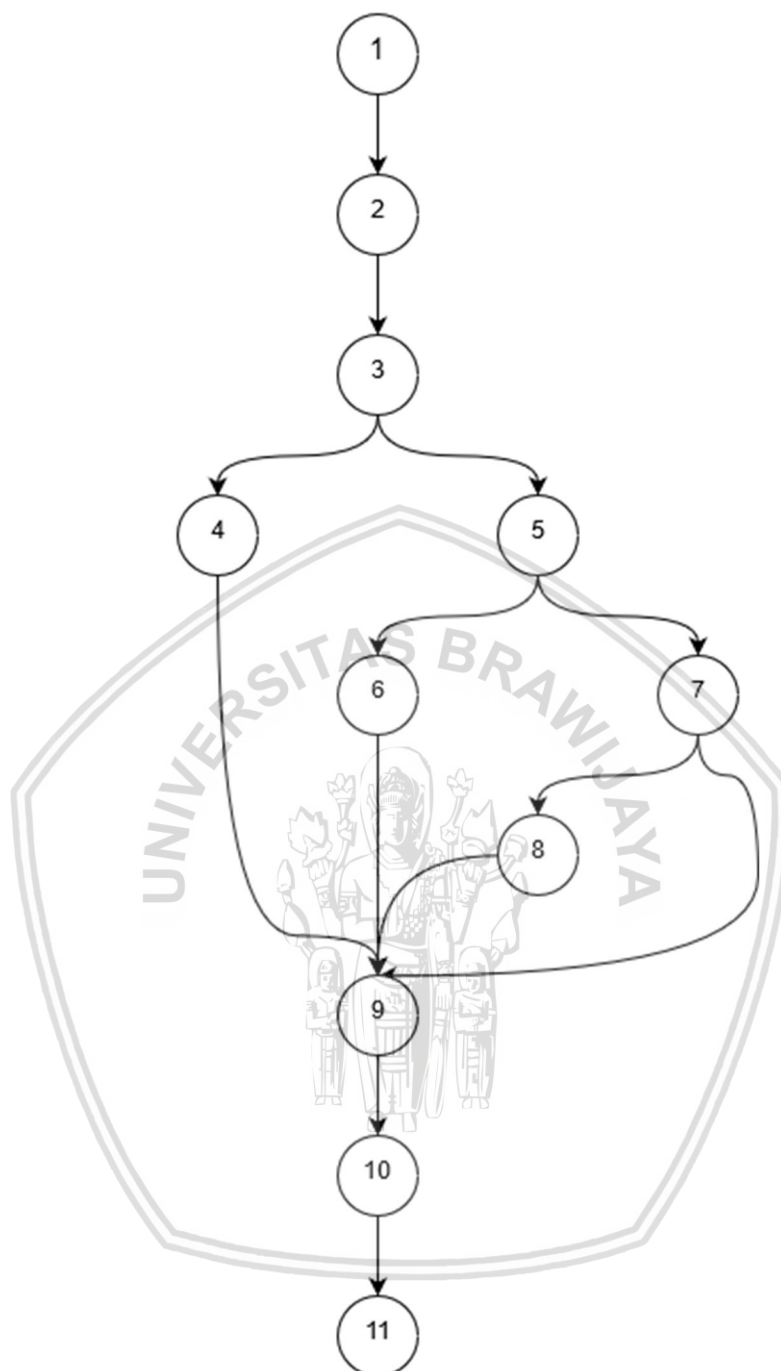
Dalam Sub bab ini akan dijelaskan hasil dari pengujian White Box dari simulasi pemadam kebakaran.

1. Prosedur Perhitungan Skor (Jumlah Bintang)



Gambar 5.1 Pseudocode Prosedur Menang

Dari pseudocode pada Gambar 5.1, didapat *flow graph* sebagai berikut.



Gambar 5.2 Flow Graph Prosedur Perhitungan Skor (jumlah bintang)

Dari *flow graph* pada Gambar 5.2, dapat ditentukan nilai kompleksitas siklomatis (*cyclomatic complexity*) dari algoritma perhitungan skor (jumlah bintang) dengan menggunakan persamaan $V(G) = E - N + 2$. Pada persamaan tersebut $V(G)$ merupakan nilai kompleksitas siklomatis, E merupakan jumlah sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*),

$$V(G) = E - N + 2$$

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 4 buah basis set dari jalur independent, yaitu :

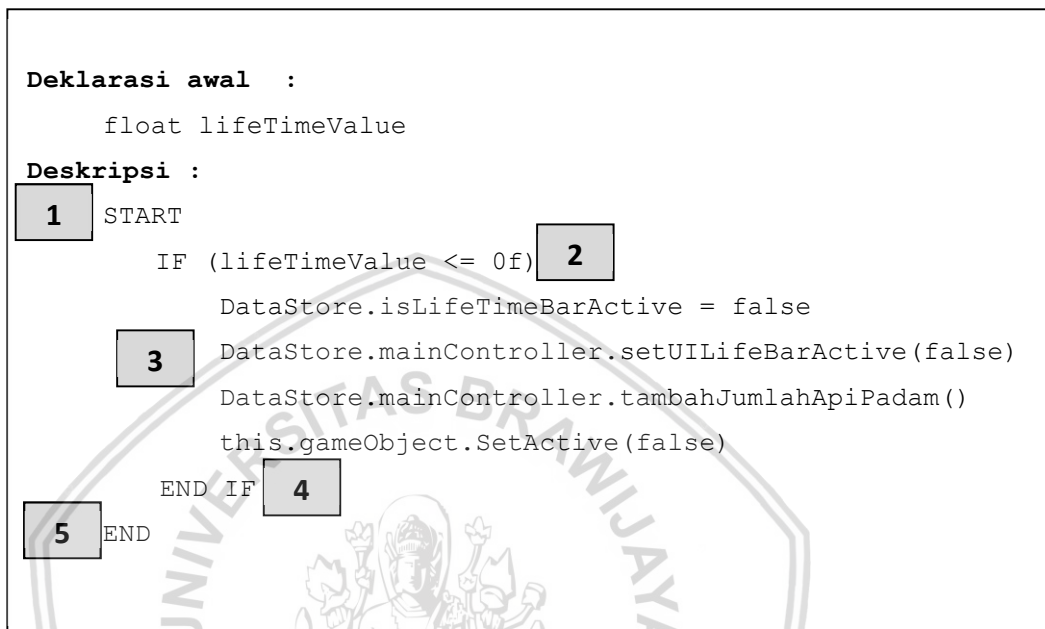
Jalur Independen:

1. 1-2-3-4-9-10-11
2. 1-2-3-5-6-9-10-11
3. 1-2-3-5-7-8-9-10-11
4. 1-2-3-5-7-9-10-11

Tabel 5.1 Kasus Uji Prosedur Perhitungan Skor (jumlah bintang)

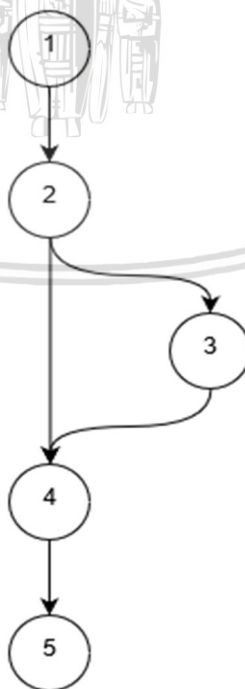
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Menyelesaikan level dengan sisa waktu lebih besar atau sama dengan 50% dari waktu total yang diberikan pada level tersebut	Muncul 3 buah bintang	Muncul 3 buah bintang
2	Menyelesaikan level dengan sisa waktu lebih besar atau sama dengan 25% sampai 49% dari waktu total yang diberikan pada level tersebut	Muncul 2 buah bintang	Muncul 2 buah bintang
3	Menyelesaikan level dengan sisa waktu lebih besar dari 0% sampai 24% dari waktu total yang diberikan pada level tersebut	Muncul 1 buah bintang	Muncul 0 buah bintang
4	Memainkan level sampai sisa waktu habis (0)	Tidak muncul bintang	Tidak muncul bintang

2. Prosedur Menghancurkan Objek Api



Gambar 5.3 Pseudocode Prosedur Menghancurkan Objek Api

Dari pseudocode pada Gambar 5.3, didapat *flow graph* sebagai berikut.



Gambar 5.4 Flow graph Menghancurkan Objek Api

Dari *flow graph* pada Gambar 5.4, dapat ditentukan nilai kompleksitas siklomatis (*cyclomatic complexity*) dari algoritma menghancurkan objek api dengan menggunakan persamaan $V(G) = E - N + 2$. Pada persamaan tersebut $V(G)$ merupakan nilai kompleksitas siklomatis, E merupakan jumlah sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*),

$$V(G) = E - N + 2$$

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 2 buah basis set dari jalur independent, yaitu :

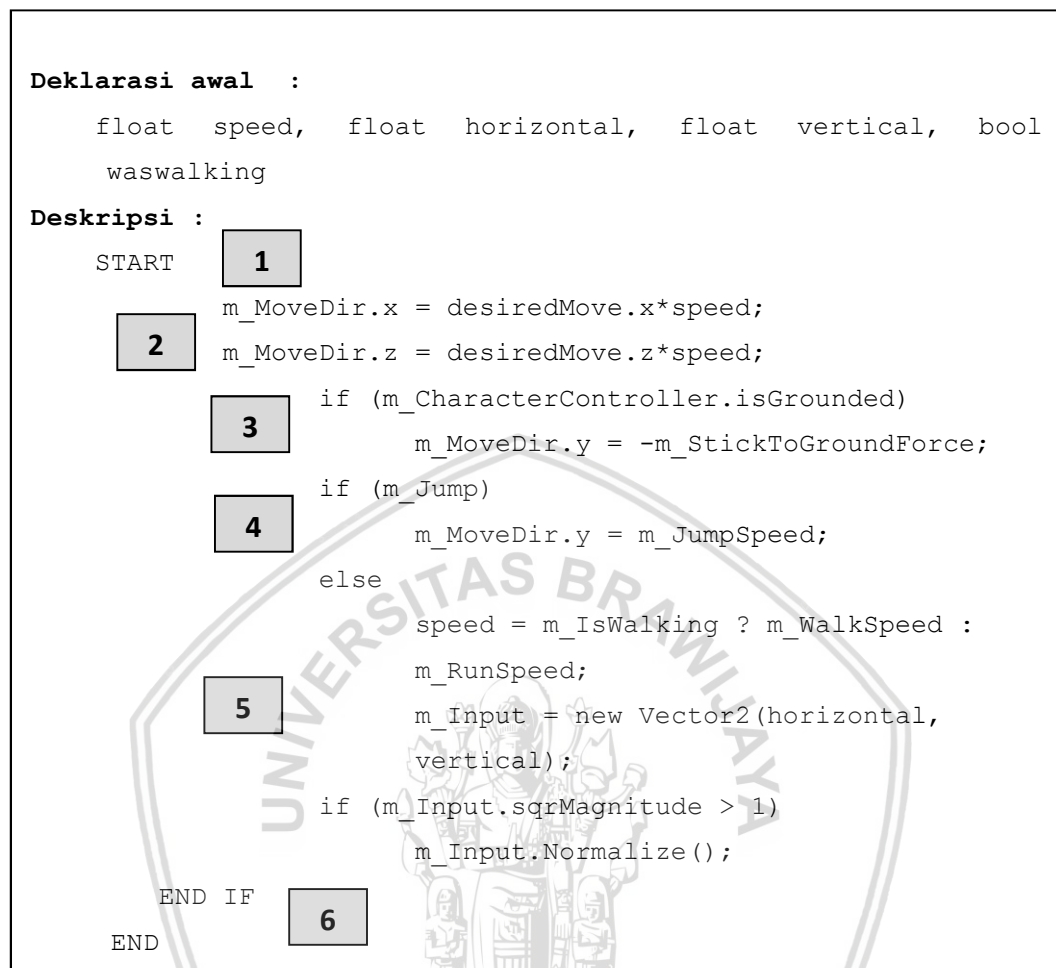
Jalur Independen:

1. 1-2-3-4-5
2. 1-2-4-5

Tabel 5.2 Tabel Kasus Uji Perhitungan Timbangan

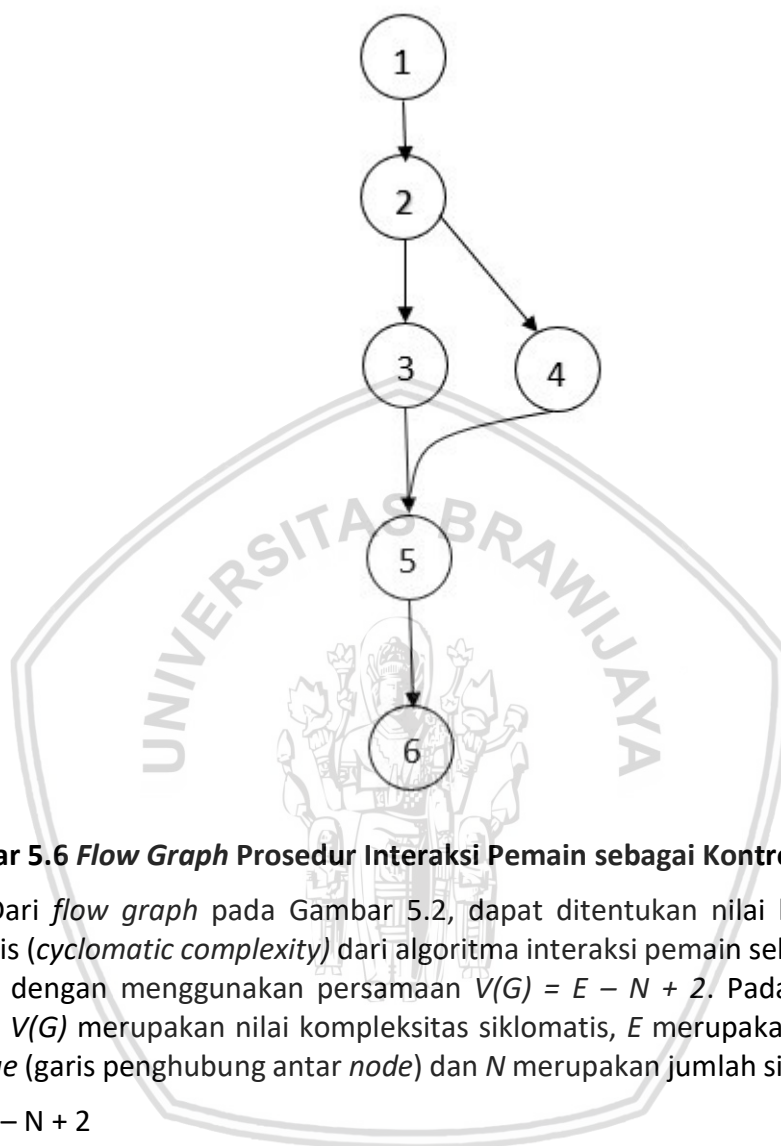
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Membuat nyawa/ <i>life time</i> dari objek api habis atau 0	Objek api hilang	Objek api hilang
2	Membiarkan nyawa/ <i>life time</i> dari objek api lebih dari 0	Objek api tetap ada	Objek api tetap ada

3. Prosedur Interaksi Pemain sebagai Kontrol Karakter



Gambar 5.5 Pseudocode Prosedur Interaksi Pemain sebagai Kontrol Karakter

Dari pseudocode pada Gambar 5.5, didapat *flow graph* sebagai berikut.



Gambar 5.6 Flow Graph Prosedur Interaksi Pemain sebagai Kontrol Karakter

Dari *flow graph* pada Gambar 5.2, dapat ditentukan nilai kompleksitas siklomatis (*cyclomatic complexity*) dari algoritma interaksi pemain sebagai kontrol karakter dengan menggunakan persamaan $V(G) = E - N + 2$. Pada persamaan tersebut $V(G)$ merupakan nilai kompleksitas siklomatis, E merupakan jumlah sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*),

$$V(G) = E - N + 2$$

$$V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 4 buah basis set dari jalur independent, yaitu :

Jalur Independen:

1. 1-2-3-5-6
2. 1-2-4-5-6

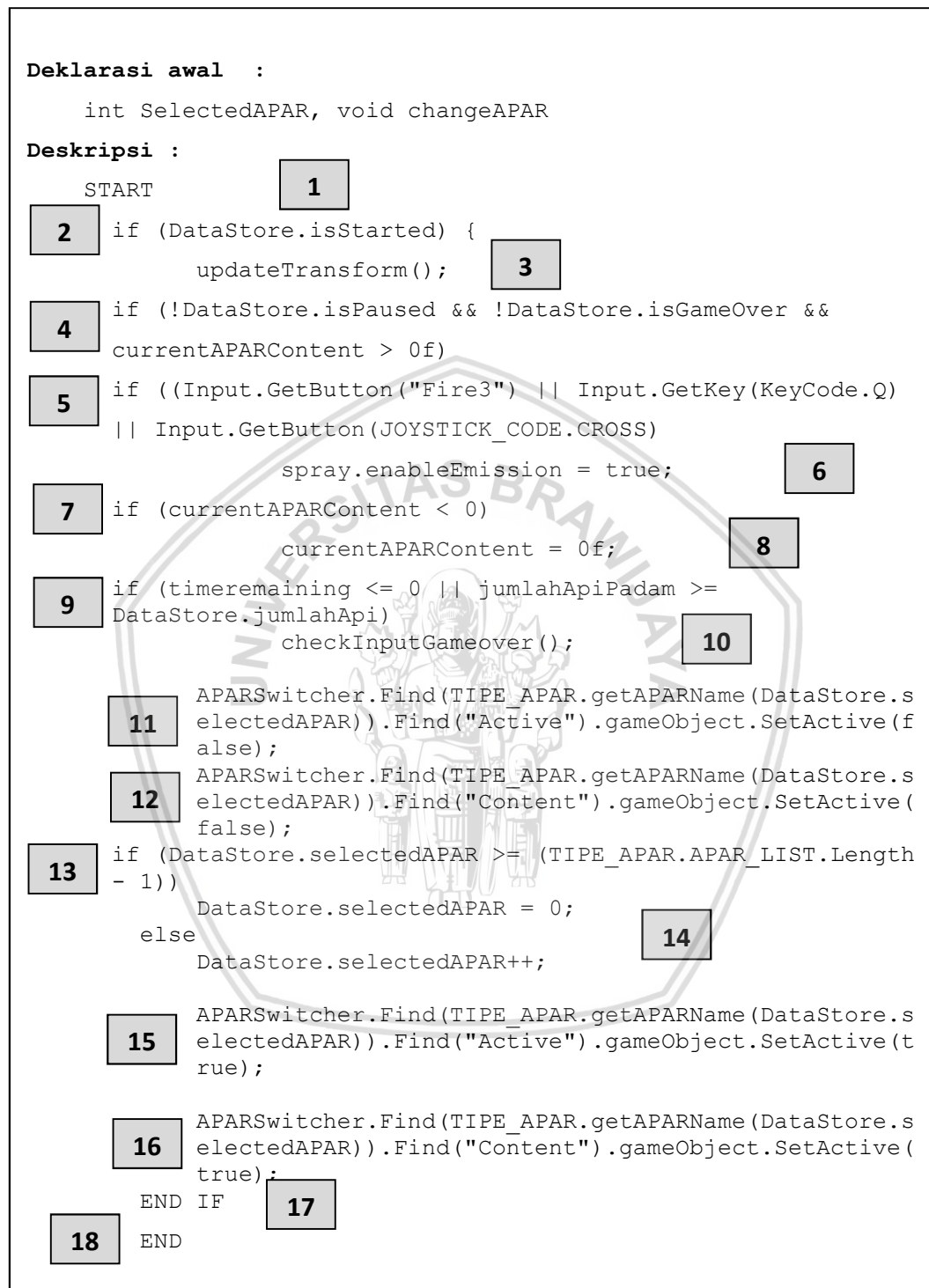
Tabel 5.3 Kasus Uji Prosedur Interaksi Pemain sebagai Kontrol Karakter

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Menjalankan karakter	Karakter berjalan sesuai tombol yang ditekan	Karakter berjalan sesuai tombol yang ditekan
2	Karakter melompat	Karakter melompat sesuai tombol yang ditekan	Karakter melompat sesuai tombol yang ditekan

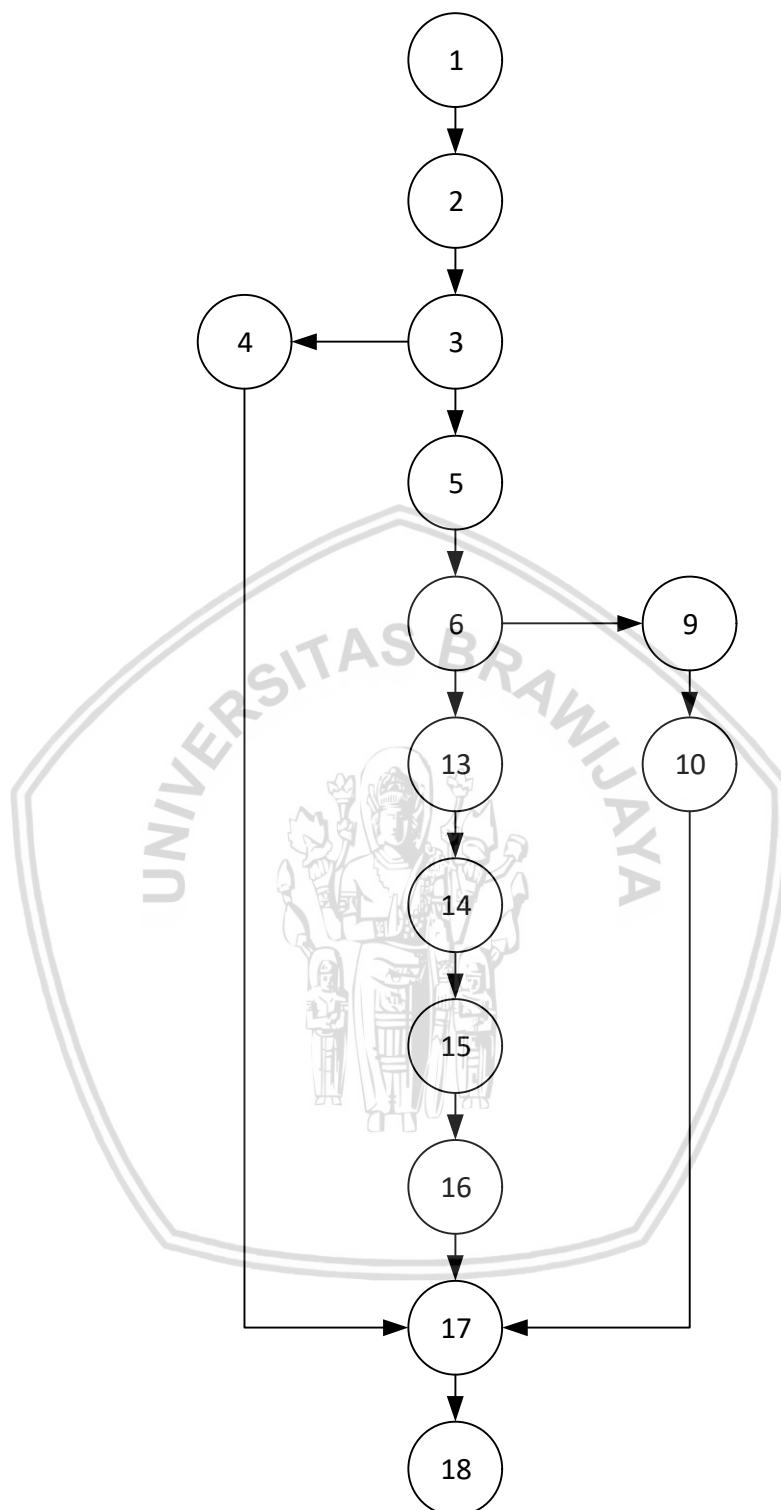
4. Prosedur mengaktifkan dan mengganti jenis semprotan APAR.



Gambar 5.7 Pseudocode Prosedur mengaktifkan dan mengganti jenis semprotan APAR.



Dari pseudocode pada Gambar 5.1, didapat *flow graph* sebagai berikut.



Gambar 5.8 Flow Graph Prosedur mengaktifkan dan mengganti jenis semprotan APAR.

Dari *flow graph* pada Gambar 5.2, dapat ditentukan nilai kompleksitas siklomatis (*cyclomatic complexity*) dari algoritma mengaktifkan dan mengganti jenis semprotan APAR. dengan menggunakan persamaan $V(G) = E - N + 2$. Pada persamaan tersebut $V(G)$ merupakan nilai kompleksitas siklomatis, E merupakan

jumlah sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*),

$$V(G) = E - N + 2$$

$$V(G) = 15 - 18 + 2$$

$$V(G) = -1$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 3 buah basis set dari jalur independent, yaitu :

Jalur Independen:

3. 1-2-3-4-17-18
4. 1-2-3-5-6-13-14-15-16-17-18
5. 1-2-3-5-6-9-10-17-18

Tabel 5.4 Kasus Uji Prosedur Mengaktifkan dan Mengganti Jenis Semprotan APAR.

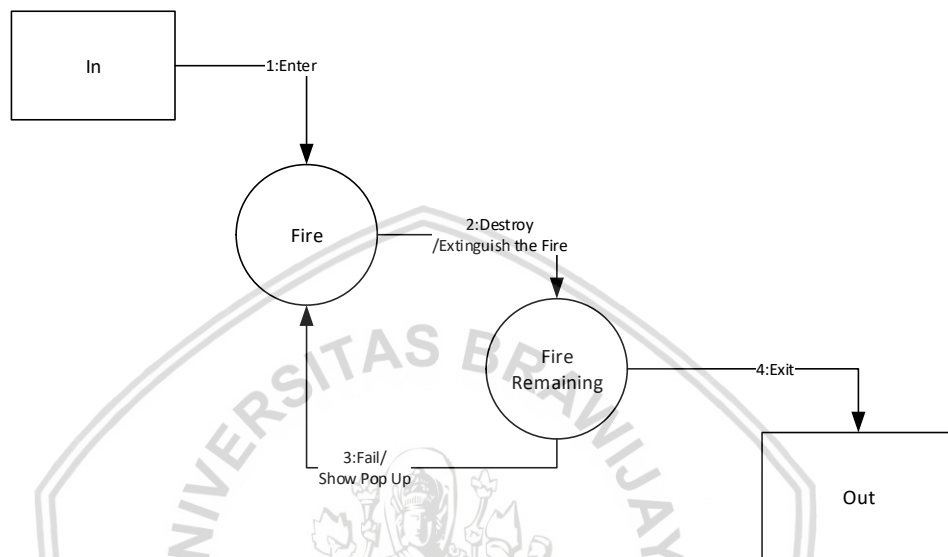
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Mengaktifkan APAR saat pause game dan game over	APAR dinonaktifkan	APAR dinonaktifkan
2	Mengaktifkan dan mengganti APAR yang masih mempunyai isi tabung	APAR dapat digunakan dan diganti	APAR dapat digunakan dan diganti
3	Mengaktifkan dan mengganti APAR yang tidak mempunyai isi tabung	APAR tidak dapat digunakan	APAR tidak dapat digunakan

5.2 Hasil Pengujian Menggunakan *Black Box*

Dalam Sub bab ini akan dijelaskan pengujian Black Box dengan menggunakan metode Test Flow Diagram (TFD). TFD digunakan untuk menggambarkan behavior dari sudut player. TFD dibuat melalui pendekatan

formal untuk testing sebuah desain. Berikut ini Pengujian TFD pada simulasi pemadam kebakaran pada gambar 5.3

Pada Gambar 5.3 dibawah ini menggambarkan pengujian *Test Flow Diagram* terhadap simulasi pemadam kebakaran.



Gambar 5.9 Pengujian Test Flow Diagram pada simulasi

Test case yang dihasilkan:

1. Terminator “In”, event: *Enter*
 - a. Menjalankan simulasi pada Smartphone.
 - b. Memilih tombol *play*.
 - c. Memilih level.
2. Event “Destroy”, action: Extinguish the Fire
 - a. Menghancurkan api untuk menyelesaikan permainan.
 - b. Jenis api yang dihancurkan disesuaikan dengan jenis APAR.
3. State “Fire Remaining”
 - a. Memeriksa apakah api masih tersisa.
4. Event “Fail”, action: Show Pop Up
 - b. Api tidak berhasil dipadamkan dan pemain harus mengulang simulasi.
5. Event “exit”

- a. Mengakhiri proses, pemain diberikan 3 pilihan yaitu kembali ke home atau mengulang level.

6. Terminator "Out"

- a. Keluar dari kondisi.

Data Dictionary:

1. Menghancurkan Objek/penghalang
 - a. Memilih jenis APAR.
 - b. Cursor APAR diarahkan ke api.
 - c. Health Bar api berkurang.
 - d. Api hancur.
2. Memilih Menu
 - a. Joystick diarahkan ke tombol menu yang dipilih.
 - b. Tunggu loading.

Tabel 5.5 Tabel Kasus Uji Pengujian Prosedur

NO	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan	Status
1	Memilih icon Play	Sistem diharapkan menampilkan halaman untuk memilih level	Sistem menampilkan halaman untuk memilih level	Valid
2	Memilih icon Level	Sistem diharapkan masuk kedalam game sesuai level yang dipilih	Sistem masuk kedalam game sesuai level yang dipilih	Valid
3	Menampilkan Tutorial	Sistem diharapkan menampilkan tutorial cara bermain	Sistem menampilkan tutorial cara bermain	Valid

4	Menggerakkan Pemain	Sistem diharapkan dapat menggerakkan kamera sesuai arah yang dipilih	Sistem dapat menggerakkan kamera sesuai arah yang dipilih	Valid
5	Memadamkan api	Sistem diharapkan dapat mengurangi lifetime dari api dengan jenis APAR yang benar	Sistem dapat mengurangi lifetime dari api dengan jenis APAR yang benar	Valid
6	Mengganti jenis APAR	Sistem diharapkan dapat mengganti jenis APAR saat pengguna menekan tombol persegi	Sistem dapat mengganti jenis APAR saat pengguna menekan tombol persegi	Valid
7	Mendapatkan bintang	Sistem diharapkan dapat menampilkan bintang sesuai dengan goal yang diselesaikan oleh pemain	Sistem dapat menampilkan bintang sesuai dengan goal yang diselesaikan oleh pemain	Valid

5.3 Pengujian Usabilitas

Pengujian ini menggunakan kuesioner untuk penilaian simulasi dan permainan pemadam kebakaran. Metode SUS atau *System Usability Scale* digunakan dalam pembuatan kuisisioner ini. Metode SUS tetap mendapatkan hasil yang dapat diandalkan walaupun sampel yang digunakan sangat kecil.

Dalam Metode ini terdapat 10 poin pernyataan yang terdiri dari 5 pernyataan negatif dan 5 pernyataan positif. Kuisisioner ini dibagikan kepada petugas pemadam kebakaran dan responden. Untuk responden akan berkaitan dengan pengujian pada simulasi dan permainan.

Daftar pernyataan dari kuesioner yang diberikan ke responden sebagai sampel yang akan diuji dapat dilihat pada Tabel 5.4

Tabel 5.6 Kuisisioner Pengujian Usabilitas Simulasi dan Permainan

No	Pernyataan	Rating Pengguna				
		Sangat tidak setuju				Sangat setuju
		1	2	3	4	5
1	Saya merasa dapat menggunakan simulasi ini dengan sangat percaya diri					
2	Saya perlu belajar banyak hal sebelum menggunakan simulasi ini					
3	Menurut saya, kontrol dan keseluruhan dari simulasi ini mudah digunakan					
4	Menurut saya, terdapat kerumitan yang tidak penting didalam simulasi ini					
5	Menurut saya, simulasi ini dapat membantu saya menggunakan APAR pada lingkungan disekitar saya					
6	Menurut saya, banyak fungsi pada simulasi ini berjalan secara tidak konsisten					
7	Menurut saya, setiap fungsi pada simulasi ini dapat berjalan dengan sangat baik					
8	Menurut saya, saya membutuhkan bantuan dari teknisi untuk menggunakan simulasi ini					
9	Menurut saya, kebanyakan orang akan belajar dengan cepat dalam menggunakan aplikasi ini					
10	Menurut saya, simulasi ini mempunyai penggunaan yang sangat rumit					

Pernyataan positif ditujukan pada pernyataan nomor ganjil, sedangkan pernyataan negative ditujukan pada pernyataan nomor genap. Kuisisioner ini akan dibagikan kepada 3 responden dan 1 orang petugas pemadam kebakaran.

5.3.1 Perhitungan Skor Kuisisioner

Didalam pernyataan terdapat skala 1 sampai 5 (sangat tidak setuju sampai sangat setuju) yang akan diisi oleh responden. Perhitungan skor akan dilakukan pada kuisisioner yang telah dibagikan. Detail perhitungan akan menggunakan rumus dibawah ini :

- Pernyataan dengan nomor ganjil, skala yang dipilih akan dikurangi 1 (skala -1)
- Pernyataan dengan nomor genap, 5 akan dikurangi skala yang telah dipilih (5 -skala)
- Untuk hasil perhitungan skor akan dihitung berdasarkan jumlah skor dari penjumlahan skor pernyataan 1 sampai 5 dengan skor pernyataan nomor 6 sampai 10 lalu dikalikan 2,5 agar didapatkan nilai dari 0 – 100.

Standar kepuasan untuk penilaian SUS yang baik adalah skor 68 (sauro, 2011). Jika skor yang didapatkan lebih dari 68 maka setiap fungsi simulasi dan permainan berjalan dengan baik. Jika skor yang didapatkan kurang dari 68 maka simulasi dan permainan tidak berjalan dengan baik.

5.4 Analisis

5.4.1 Analisis Pengujian Dengan *White Box Testing*

Hasil analisis didapatkan dari melihat kecocokan antara perancangan simulasi yang telah dirancang dengan fungsi dari implementasi unit modul. Dari seluruh pengujian unit yang dilakukan, jumlah jalur pada logika setiap method telah sesuai dengan perhitungan kompleksitas siklomatis. Kasus uji yang dibuat telah sesuai berdasarkan jumlah jalur yang telah diuji dan hasil yang diharapkan juga sesuai dengan hal tersebut. Maka dapat diambil kesimpulan bahwa unit modul sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

5.4.2 Analisis Pengujian Dengan *Black Box Testing*

Berdasarkan hasil uji kesesuaian hasil implementasi dengan fungsionalitas prosedur simulasi pemadam kebakaran mendapatkan hasil yang sesuai dengan kebutuhan simulasi ini. Dari analisa hasil pengujian didapatkan bahwa implementasi dan fungsionalitas simulasi ini sudah memenuhi kebutuhan.

5.4.3 Hasil Pengujian Usabilitas

Pengujian dilakukan pada setiap responden yang diberikan kuesioner dan wajib menjawab setiap pernyataan. Pada Tabel 5.5 merupakan salah satu sampel dari pengisian kuesioner yang dilakukan oleh responden.

Tabel 5.7 Sampel Perhitungan dari Pengisian Kuisisioner

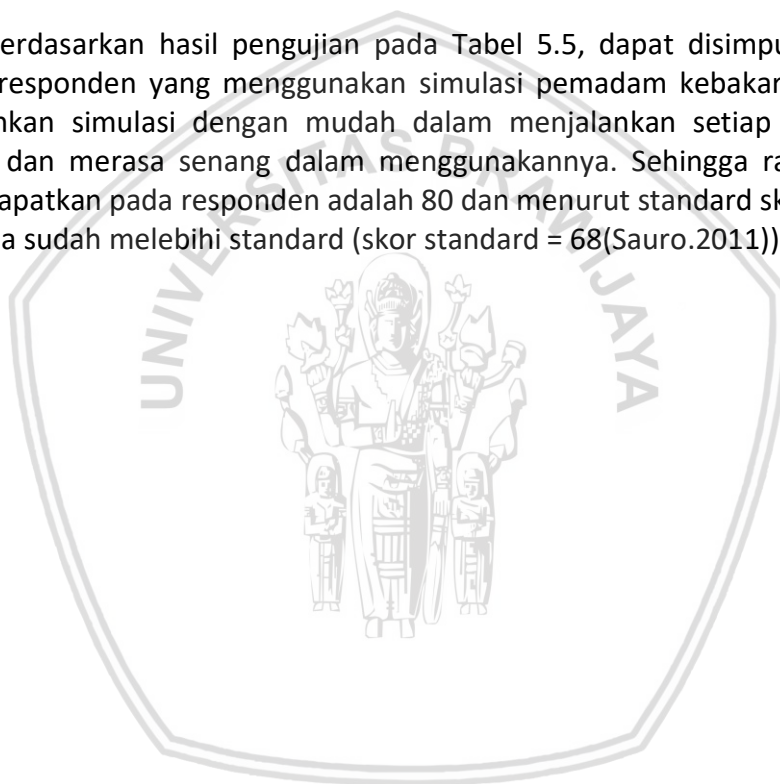
No.	Pernyataan	Penilaian oleh pengguna	Perhitungan	Hasil
1	Saya merasa dapat menggunakan simulasi ini dengan sangat percaya diri	4	4-1	3
2	Saya perlu belajar banyak hal sebelum menggunakan simulasi ini	1	5-1	4
3	Menurut saya, kontrol dan keseluruhan dari simulasi ini mudah digunakan	5	5-1	4
4	Menurut saya, terdapat kerumitan yang tidak penting didalam simulasi ini	1	5-1	4
5	Menurut saya, simulasi ini dapat membantu saya menggunakan APAR pada lingkungan disekitar saya	4	4-1	3
6	Menurut saya, banyak fungsi pada simulasi ini berjalan secara tidak konsisten	1	5-1	4
7	Menurut saya, setiap fungsi pada simulasi ini dapat berjalan dengan sangat baik	5	5-1	4
8	Menurut saya, saya membutuhkan bantuan dari teknisi untuk menggunakan simulasi ini	2	5-2	3
9	Menurut saya, kebanyakan orang akan belajar dengan cepat dalam menggunakan aplikasi ini	4	4-1	3
10	Menurut saya, simulasi ini mempunyai penggunaan yang sangat rumit	3	5-3	2
	TOTAL		Jumlah *2,5	85

Seluruh data pengisian kuisioner oleh pengguna dapat dilihat pada lampiran. Kemudia Tabel 5.6 merupakan hasil perhitungan kuesioner yang dibagikan kepada 3 responden.

Tabel 5.6 Hasil Skor Pengujian Usabilitas Pada Pengguna

Nomor Responden	Total Skor
Responden 1	85
Responden 2	82,5
Responden 3	72,5
Rata-rata Total Skor	80

Berdasarkan hasil pengujian pada Tabel 5.5, dapat disimpulkan bahwa seluruh responden yang menggunakan simulasi pemadam kebakaran ini dapat menjalankan simulasi dengan mudah dalam menjalankan setiap fungsi pada simulasi dan merasa senang dalam menggunakannya. Sehingga rata-rata skor yang didapatkan pada responden adalah 80 dan menurut standard skor kepuasan pengguna sudah melebihi standard (skor standard = 68(Sauro.2011)).



BAB 6 KESIMPULAN

6.1 Kesimpulan

Dari hasil pengujian yang dilakukan terhadap simulasi pemadam kebakaran didapatkan beberapa kesimpulan, yaitu:

1. Pertama penulis mengumpulkan kebutuhan fungsional yang berasal dari hasil wawancara dengan Kepala UPT Pemadam Kebakaran kota Malang. Dari kebutuhan tersebut penulis merancang permainan menggunakan *MDA framework* yang dimulai dengan menentukan elemen formal dari permainan yang dibuat. Permainan yang sudah dirancang diimplementasikan ke dalam *virtual reality* menggunakan Google VR sdk untuk unity versi 0.9.1.
2. Berdasarkan hasil menggunakan *white box testing*, didapatkan jumlah jalur pada logika setiap method dan prosedur telah sesuai dengan perhitungan kompleksitas siklomatis dan setiap jalur independen dengan hasil pengujian sesuai dengan hasil yang diharapkan. Sedangkan Berdasarkan hasil menggunakan metode *black box* dengan melakukan pengujian test flow diagram menggunakan teknik jalur minimum path generation, didapatkan hasil pengujian sesuai dengan yang diharapkan tanpa adanya bug dan tidak ditemukannya kesalahan fungsional.
3. Berdasarkan hasil pengujian usabilitas terhadap pengguna, Pengujian usabilitas menggunakan kuesioner dengan metode SUS dan dibagikan kepada 3 responden. dapat disimpulkan dari jumlah skor rata-rata yaitu 80 dan menurut standard skor kepuasan pengguna sudah melebihi standard (skor standard = 68(Sauro.2011)) bahwa simulasi dapat digunakan dengan mudah dan berjalan dengan baik.

6.2 Saran

Saran untuk pengembangan simulasi pemadam kebakaran lebih lanjut antara lain:

1. Dapat dilakukan evaluasi pembelajaran untuk pengenalan APAR terhadap pengguna.
2. Perlunya pengembangan pada controller, agar pemain dapat menggunakan controller yang lebih menunjang indra saat menggunakan virtual reality.

DAFTAR PUSTAKA

- Michael Heim. 1993. *The Metaphysics of Virtual Reality*, Oxford University Press. Tersedia di <<http://www.mheim.com/wp-content/uploads/2014/05/The-Metaphysics-of-Virtual-Reality-1991-Meckler.pdf>> [Diakses 20 Agustus 2017]
- Andy Beane. 2012. *3D Animation Essentials*. Tersedia di <<https://www.wiley.com/WileyCDA/WileyTitle/productCd-1118147480,miniSiteCd-SYBEX.html>> [Diakses 23 Agustus 2017]
- Teguh Hambudi. 2015. #1 Professional General Affair Panduan Bagian Umum Perusahaan Modern, Visimedia.
- Maria T Schultheis, Albert A Rizo. 2001. *The Application of Virtual Reality Technology in Rehabilitation, the Educational Publishing Foundation*. Tersedia di <<http://doi.apa.org/journals/rep/46/3/296.pdf>> [Diakses 20 Agustus 2017]
- Pressman R S. 2001. *Software Engineering: A Practitioner's Approach, 7th Edition*. Mc Graw Hill. Tersedia di <http://dinus.ac.id/repository/docs/ajar/RPL-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_.pdf> [Diakses 21 Agustus 2017]
- Roedavan R. 2014. *UNITY Tutorial Game Engine*. Bandung: Informatika.
- Direktorat Jenderal Pembinaan Pelatihan dan Produktivitas. 2009. Materi Pelatihan Berbasis Kompetensi Sektor Pemadam Kebakaran. Jakarta Selatan.
- Rogers S. 2010. *Level Up! The Guide to Great Video Game Design*. US. Willey. Tersedia di <<https://www.wiley.com/en-us/Level+Up%21+The+Guide+to+Great+Video+Game+Design%2C+2nd+Edition-p-9781118877166>> [Diakses 19 Agustus 2017]
- Robin Hunicke, M. L. R. Z. 2015. MDA: A Formal Approach to *Game Design and Game Research*. *Game Developers Conference*. Tersedia di <https://www.researchgate.net/publication/228884866_MDA_A_Formal_Approach_to_Game_Design_and_Game_Research> [Diakses 20 Agustus 2017]
- Hejlsberg A, Torgersen M, Wiltamuth S, Golde P. 2011. *The C# Programming Language (Covering C# 4.0)*. Addison Wesley Profesional. Tersedia di <<https://dl.acm.org/citation.cfm?id=1502323>> [Diakses 19 Agustus 2017]
- Sauro J, . 2011, February 2. *Measuring Usability With The System Usability Scale (SUS)*. Tersedia di <<https://measuringu.com/sus/>> [Diakses 20 Agustus 2017]

- Peres, S & Pham. 2013. *Validation of the System Usability Scale (SUS). Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Tersedia di <https://www.researchgate.net/publication/273297038_Validation_of_the_System_Usability_Scale_SUS> [Diakses 23 Agustus 2017]
- Alexander J. 2016. *Developing Virtual Reality Simulations for Office-Based Medical Emergencies*. Tersedia di <<https://journals.tdl.org/jvwr/index.php/jvwr/article/view/7184>> [Diakses 2 Agustus 2018]

